

SAT 380.

N94-13673

Unclas

G3/39 0183140

FINAL TECHNICAL REPORT

ON

AN EXPERT SYSTEM FOR INTEGRATED STRUCTURAL ANALYSIS AND DESIGN OPTIMIZATION FOR AEROSPACE STRUCTURES

SBIR 1987 Phase II

Prepared for

NASA--Lewis Research Center
Cleveland, Ohio 44135

(NASA-CR-194225) AN EXPERT SYSTEM
FOR INTEGRATED STRUCTURAL ANALYSIS
AND DESIGN OPTIMIZATION FOR
AEROSPACE STRUCTURES Final
Technical Report (Structural
Analysis Technology) 70 p

This material is based upon work supported by the National Aeronautics and Space Administration (NASA) under Contract Number NAS3-25642. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the author(s), and do not necessarily reflect the views of NASA.

SBIR - -0404-1319

Release Date - 7-27-93

Prepared by

Structural Analysis Technologies, Inc.
4677 Old Ironsides Drive, Suite 250
Santa Clara, CA 95054
Tel: (408)496-1120 Fax: (408)496-1196

April 1992

TABLE OF CONTENTS

Acknowledgements

Executive Summary.....S - 1

1. Introduction..... 1 - 1

2. Review of the State-of-the-Art in Expert Systems
Technology and Integrated Engineering Systems..... 2 - 1

3. Functional Specifications and Basic Architecture
of the Expert Systems..... 3 - 1

4. Object-Oriented Knowledge Representation Scheme..... 4 - 1

5. AutoDesign™ and Its Expert System Modules..... 5 - 1

6. Conclusions..... 6 - 1

References..... R - 1

ACKNOWLEDGEMENTS

This research work was supported by National Aeronautics and Space Administration--Lewis Research Center, under Contract Number NAS3-25642. The NASA Project Manager was Dr. Laszlo Berke. We gratefully acknowledge many valuable suggestions, technical references, and general guidance provided by Dr. Berke throughout the course of this project.

EXECUTIVE SUMMARY

This report presents the results of a research study on the development of an expert system for integrated structural analysis and design optimization. An Object Representation Language (ORL) was developed first in conjunction with a rule-based system. This ORL/AI shell was then used to develop expert systems to provide assistance with a variety of structural analysis and design optimization tasks, in conjunction with procedural modules for finite element structural analysis and design optimization.

The main goal of the research study was to provide expertise, judgment and reasoning capabilities in the aerospace structural design process. This will allow engineers performing structural analysis and design, even without extensive experience in the field, to develop error-free, efficient and reliable structural designs very rapidly and cost-effectively. This would not only improve the productivity of design engineers and analysts, but also significantly reduce time to completion of structural design.

An extensive literature survey in the field of structural analysis, design optimization, artificial intelligence and database management systems and their application to the structural design process was first performed. A feasibility study was then performed, and the architecture and the conceptual design for the integrated "intelligent" structural analysis and design optimization software was then developed.

An Object Representation Language (ORL), in conjunction with a rule-based system, was then developed using C++. Such an approach would improve the expressiveness for knowledge representation (especially for structural analysis and design applications), provide ability to build very large and practical expert systems, and provide an efficient way for storing knowledge. Functional specifications for the expert systems were then developed. The ORL/AI shell was then used to develop a variety of modules of expert systems for a variety of modeling, finite element analysis and design optimization tasks in the integrated aerospace structural design process. These expert systems were developed to work in conjunction with procedural finite element structural analysis and design optimization modules (developed in-house at SAT, Inc.). The complete software, AutoDesign™, so developed, can be used for integrated "intelligent" structural analysis and design optimization.

The software has been beta-tested at a variety of companies, used by a range of engineers with different levels of background and expertise. Based on the feedback obtained by such users, conclusions have been developed and provided in this report.

1. INTRODUCTION

This report presents the results of a research study on the development of a shell consisting of an Object Representation Language (ORL), in conjunction with a rule-based system, and expert system modules, developed using this ORL/shell for integrated structural analysis and design optimization of aerospace structures. These expert system modules were developed in conjunction with a procedural program, AutoDesign™ (developed in-house at SAT, Inc.), for structural analysis and design optimization. These expert system modules would provide assistance to the user in developing a finite element model, performing structural analysis, selecting design variables, selecting type and location of constraints, and deciding which optimization procedure to use, etc.

Thus, the integrated software, AutoDesign™, provides the capabilities for structural analysis and design optimization, while at the same time, the associated expert system modules mimic the experts so that the user is allowed to make better analysis/design decisions; thus improving his productivity and reducing time to completion, and introduction to market, of products and systems.

The structural design process involves a series of iterative analyses and designs, including structural optimization, starting from conceptual design all the way to the detailed final design and verification analysis. Impressive developments have taken place in the last two decades in the areas of finite element structural analysis and design optimization with numerous major, sophisticated computer programs available for a range of structural analyses (linear, nonlinear, dynamic, buckling, etc.), as well as for structural optimization (linear and nonlinear mathematical programming and optimality criteria, etc.). However, no general purpose integrated capabilities exist that would start from a conceptual/preliminary design with approximate member sizes and would automatically develop the final detailed optimum design. Also, commonly used software packages for structural analysis and design/optimization are extremely cumbersome to use without expert training and judgement, are subject to errors and unknown accuracy limitations, are very rigid in their structure and flow, and are not geared to answer the "what if" questions that a designer would like to ask to modify or optimize his design. It was therefore clear that an integrated design package was needed which could not only provide the latest capabilities in stress analysis and design optimization, but could also integrate the techniques of structural analysis and design optimization with artificial intelligence and expert systems technology to create "An Intelligent Automated Integrated Design Capability".

In the Phase I research and development effort of this project, an architecture and conceptual design for such an integrated software was developed, and its feasibility was investigated.

In the Phase II research and development effort, described in this report, a detailed development of such an expert system and its components was carried out. Chapter 2 of this report presents a review of the state-of-the-art in expert systems technology and integrated engineering systems. Chapter 3 of this report presents the functional specifications and the architecture of the system as suggested in Phase I, followed by a description of the Object Representation Language (ORL) and the rule-based shell in Chapter 4. Chapter 5 presents the expert system modules developed using the ORL/AI shell described in Chapter 4, in conjunction with AutoDesign™, the procedural software (developed in-house at SAT) for integrated finite element structural analysis and design optimization. Finally, conclusions are included in Chapter 6. References are then presented.

2. REVIEW OF THE STATE-OF-THE-ART IN EXPERT SYSTEMS TECHNOLOGY AND INTEGRATED ENGINEERING SYSTEMS

The engineering design process involves a number of tasks requiring different types of technologies, expertise and processing. Since the advent of the computer, continuous attempts have been made to automate these tasks and develop computer-aided tools to assist with their performance.

A very significant level of progress has occurred in the last two decades in developing computer-aided design tools. In the field of structural analysis, the principal area of progress has been the development of finite element programs, such as, NASTRAN, SAP, STRUDL, STARDYN, SPAR/EAL, ANSYS, MARC, ADINA, ABACUS and MHOST, etc., that permit the stress analysis of a large number of complex structures, subject to a variety of static, thermal and dynamic loadings, in linear and nonlinear (e.g., ANSYS, MARC, ADINA, ABACUS and MHOST) regimes.

The finite element analysis methods, and the codes based on these methods such as those mentioned above, however, provide only an analysis for the verification of a trial structure. Member sizes have to be determined by other methods. Structural optimization theory has been developed to a sophisticated level, both using linear and nonlinear mathematical programming and optimality criteria approaches. In some special cases, analysis and optimization techniques have been integrated into a single automated optimum design capability. Several design optimization packages have been developed, especially in the last decade, some of which have been used in conjunction with the above finite element analysis packages. They include, for example, ADS, MICRO-DOT, IDESIGN, CONMIN, NEWSUMT, OPTSTATE, GRG2 and ASTROS.

Other computer-aided design tools have also been developed in areas that are generally prone to the development of algorithms or procedures, which can be mechanically performed using the computer, e.g., graphics, data querying and support.

The process of engineering design involves a number of steps that can not be easily broken down into algorithms or procedures. Many researchers have studied the non-algorithmic nature of the design process (Refs. 1-4). In these studies, researchers have focussed on issues such as: the process of design; how designers think; whether design can be fully automated; etc. It is clear that engineering design is an ill-structured problem, requiring judgment, creativity, cultural conditioning, heuristic reasoning and the manipulation of large amounts of relevant and partially-relevant data from which complex inferences must be derived.

With the growth of concepts and techniques in Artificial Intelligence (AI) and significant improvements in hardware architecture and speed, the development of intelligent software for engineering design is receiving increasing attention. As of now, sufficient work has been undertaken in the area of Knowledge-Based Expert Systems (KBES's) and the architecture of integrated expert software to allow a meaningful synthesis of existing AI elements in developing a functioning integrated structural analysis and design optimization package for aerospace structures.

This chapter provides a review of the concepts and tools (relevant to this effort) that exist in the engineering design field; and, in particular, in the structural analysis and design field. In Chapter 4, the concepts and architectural elements of engineering KBES's are synthesized into a knowledge representation scheme for development of expert systems for the integrated software package for structural analysis and design optimization of aerospace structures.

Furthermore, since for integrated engineering systems, extensive and distributed data base management systems (DBMS's) are required, some research work has also been carried out recently in integrating expert systems and procedural programs with DBMS's. This has been done by developing intelligent DBMS front-ends, as well as coupling expert systems with data Bases. The integrated structural analysis and design optimization package for aerospace structures, discussed in this report, uses components and concepts from the following two categories:

1. Conventional (Procedural) Engineering Tools, e.g., algorithmic software packages that perform a given repetitive engineering function, for example for finite element structural analysis or design optimization.
2. Knowledge-Based Tools or Expert Systems that perform judgmental tasks (inferencing in complex environments), synthesize expertise, and test hypotheses, etc.

The state-of-the-art in the above technical areas is briefly discussed in the following sections. For details, the list of references at the end of this report can be referred to.

2.1 CONVENTIONAL ENGINEERING TOOLS

The conventional, procedural, engineering tools used in the structural design process are very briefly outlined in this section. Since much is known about this kind of software, and since conventional software programs are not the

primary focus of this effort (except as building blocks for the integrated system), the level of details for the description of these tools is minimal.

A. Structural Analysis Programs

As mentioned in the previous section, a considerable amount of progress has occurred in the development of structural analysis programs and techniques. Finite element packages, such as SAP, STRUDL, STARDYN, SPAR/EAL, NASTRAN, ANSYS, MARC, ADINA, ABACUS and MHOST, etc., can perform structural (stress) analysis for:

- Linear and Nonlinear Static Analyses for two-dimensional and three-dimensional arbitrary shapes and geometries
- Analyses for a large range of loadings, including vibration, thermal, pressure, shock, flutter, steady and unsteady aerodynamics, cyclic loadings (fatigue/fracture), and impact
- A variety of material types, including anisotropic materials, composites, plastics, materials with a variety of nonlinear constitutive relations

B. Design Optimization Programs

Numerous design optimization packages have been developed, especially in the last decade, some of which have been used in conjunction with the above finite element analysis packages. They include, for example, ADS, MICRO-DOT, IDESIGN, CONMIN, GRG2, NEWSUMT, OPTSTATE, OPDES-BYU, and ASTROS, and can perform optimization using the following procedures:

- Optimality Criteria Methods
- Mathematical Programming Methods, e.g., linear programming, nonlinear programming using methods of feasible directions, Newton's Methods, Generalized Reduced Gradient Methods, Quadratic Programming Method, Conjugate Gradient Method, Cost Function Bounding Method, etc.
- Hybrid Methods
- Approximate Techniques

C. Sensitivity Analysis Procedures

Several sensitivity analysis procedures are available for coupling the finite element analysis programs with design optimization programs using "direct

methods" as well as "Dummy load Methods" and "Material Derivative Methods."

D. CAD Packages and Pre-processors

Many excellent programs are now available on Work Stations and PC's for assisting with the design-drafting and graphical two and three-dimensional modeling of structures. Programs such as AUTOCAD and PATRAN produce an important productivity link in the aerospace design process — by allowing the designer to view the geometry and shape of the structure, as well as in facilitating the finite element model development of structures for analysis.

E. General Tools

General purpose analysis and computational tools, relevant to the aerospace structural design process, include the following:

- Engineering Data Bases (e.g., standard components, materials, codes and standards)
- Component Design Programs
- Probabilistic/Stochastic tools

2.2 KNOWLEDGE-BASED TOOLS

In the previous section, several computer programs, available for different steps of the structural design process, were described. These programs provide tools for solving a wide range of structural engineering problems. However, these tools are algorithmic in nature, and are not able to solve, efficiently, many problems that require engineering judgement. Furthermore, many of these programs were developed by different organizations, and no consistent format is available for exchange of information between these programs. The emerging technology of knowledge-based expert systems (KBES's), along with traditional CAD programs, offers a methodology to overcome some of the above barriers. The examples of SACON (Ref. 5), HIRISE (Ref. 6) and ALLRISE (Ref. 7) have paved the way for more research on the use of KBES for such applications.

2.2.1 Knowledge Based Expert Systems (KBES) Technology

The technology of the KBES is extensively utilized in the design of the proposed integrated structural analysis and design package for aerospace

structures. A review of this technology and the components of modern day designs of KBES's are described below.

A Knowledge-Based Expert System (KBES) is an interactive computer program package that incorporates judgement, experience, rules of thumb, and intuition acting upon a potentially large amount of domain data or knowledge to solve ill-defined, non-procedural problems. In this way, it mimics the actions and reasoning processes of an expert in its domain.

A schematic view of a typical Knowledge-Based Expert System is illustrated in Figure 2-1, and consists of the following components.

A. Knowledge Base

The Knowledge Base consists of domain-specific data, general facts and heuristics (rules of thumb) that are pertinent to the expert reasoning and problem solving performed by the KBES.

The design and implementation of the Knowledge Base is a key parameter that controls the efficiency of a KBES. A great deal of research has been performed and is continuing in the development of effective knowledge representation schemes (Refs. 10-13).

A number of formalisms, such as production rules, frames, semantic nets and object-oriented environments are available for representing knowledge. The production rule representation has been extensively used in current KBES designs. In this approach, knowledge is represented as "IF—THEN" rules or "premise—action" pairs: the action is taken if the "premise" evaluates to be true. Uncertainty in the knowledge can also be represented by means of confidence factors (Ref. 14). Other forms of representations commonly used are logic, frame-based, and object-oriented schemes.

In their most general level of complexity, the production rules can handle the following:

- Fuzzy or imprecise knowledge, using probabilistic constructs
- Redundant or contradictory rules
- Lack of knowledge in certain areas of the inferencing
- Meta rules-or rules governing the generation and firing of other rules. Meta rules are essential in the design of "Self Learning" systems, e.g., systems that can modify their own rules as more knowledge usage comes into being.

B. Knowledge Acquisition Facility

Sometimes, attached to the Knowledge Base is a Knowledge Acquisition Facility. This facility permits the continual generation of new or modified knowledge that is pertinent to the expertise of the KBES. Thus the expert system, like an expert in the field, is able to remain current, reflecting the latest body of knowledge, consensus opinions, related projects, data bases, etc.

C. Context

The context is a collection of symbols or facts that reflects the current state of the problem at hand. It consists of all the information generated during a particular program execution.

The "awareness" of the context by the expert system allows it to ask only pertinent questions and seek relevant data. The user interface can also be made greatly user friendly by utilizing context-specific querying and user responses.

D. Inference Mechanism

The Inference Mechanism (Inference Machine and Inference Engine are other terms commonly used, instead of Inference Mechanism) monitors the execution and performs the reasoning to arrive at decisions and other control actions. Various strategies for inferencing to arrive at valid conclusions or decisions exist—e.g., forward/backward chaining, unification, means end analysis, least commitment principle, reasonings by analogy, etc. A detailed description of these strategies can be found in References 15-17.

Different inferencing strategies are suitable for different expert domains. Most KBES designs, proposed for limited domain applications, provide a common Inference Mechanism for the entire software package.

E. Explanation Facility

An important aspect of an expert system is the ability to explain how it arrived at certain decisions or conclusions. In this way, the non-expert user can gain insight into the logical process utilized by a domain expert in performing project tasks. In due time, the user can be trained in using an expert system with an explanation facility and can also modify the decision process if he has more specific or detailed knowledge than the expert system.

F. User Interface

The User Interface is an important aspect of an efficient, interactive, expert system. The function of a user interface is to shield the user from having to interact with the software at an internal computer hardware/software design level. Instead, the user interacts with the software, using the following facilities:

- Windows and Pop-up menus
- Graphics devices wherever feasible
- English-like constructs

The user need not know the names of the data bases, program modules, file names, etc., that the software uses. An efficient User Interface also provides "Help" levels and diagnostics that make the program easy to learn and use. Although a significant amount of work is in progress in developing natural language interfaces (Refs. 18-19), the problem is complex and much remains to be done. The development of a natural language interface is not an objective of this effort.

G. "Blackboard" Architecture

A general framework—the "Blackboard" Architecture—for integrating knowledge from several sources—has been successfully designed and implemented (Refs. 20-21).

A "Blackboard" system consists of a number of knowledge sources that communicate through a "Blackboard" of a global data base. These knowledge sources are controlled by an Inference Mechanism, as shown in Figure 2-2.

The data that goes onto a "Blackboard" can be divided up using many different types of schemas—the most commonly proposed schema for engineering design being a multi-level data organization where each level contains a higher level of abstraction (or the next level of completed decision) based on the previous level.

The KBES components described above represent very powerful knowledge-oriented tools. These tools with modifications and additions are the basic building blocks of the proposed integrated expert software for structural analysis and design optimization. The architecture of the proposed expert software package is described in detail in Chapter 4. Examples of selected relevant knowledge-based expert systems, developed or in development (available in the literature) are presented below.

2.2.2 Examples of Selected Relevant Expert Systems

2.2.2.1 SACON

SACON (Structural Analysis Consultant) assists users of the MARC analysis package. by querying the user as to the nature of the object under design, (e.g, "is the substructure thin-walled or solid?", "is heating or cooling response of interest?"). SACON suggests an appropriate analysis strategy. This strategy consists of an analysis class (e.g., general inelastic, buckling, nonlinear-crack-growth, etc.) together with a set of recommendations of MARC features which should be activated when performing the analysis. Such recommendations include:

"activate incremental stress - incremental strain analysis,"
"model nonlinear stress-strain relation of material,"
"cumulative strain damage should be calculated."

The SACON research actually served a dual purpose. In addition to providing a more intelligent front end for MARC users, it demonstrated the domain-independence of the EMYCIN expert system environment. Based on MYCIN (Ref. 6), a system designed to assist physicians in diagnosing infections, EMYCIN employs a backward-chaining approach to problem solving. Starting with the set of possible alternatives, the system queries the user for evidence to confirm or deny each potential solution. This strategy allows SACON to provide a reasonable explanation facility. Confronted with a response of "why?" to a particular inquiry, SACON can explain its current question by paraphrasing the set of hypotheses for which it is attempting to gather evidence.

2.2.2.2 HIGHRISE and DESTINY

These complementary knowledge-based expert systems, developed at Carnegie-Mellon University, provide assistance in conceptual design and synthesis of building structures. Given the topology and geometry of the building, the expert systems provide a preliminary choice of the structural system and elements of the super-structure. They are rule-based expert systems, with data objects defined in frames in SRL.

2.2.2.3 STRUTEX

This is a prototype expert system, developed at NASA-Langley Research Center, to initially configure a structure to support point loads in two dimensions (Ref. 52). The system combines numerical and symbolic processing by the computer with interactive problem solving, aided by the

vision of the users, integrating a knowledge-based interface and inference engine, a data base interface, and graphics while keeping the knowledge-base and data base files separate. The system writes a file which can be input into a structural synthesis system which may be utilized for design.

2.2.2.4 EXADS

This is a prototype expert system, developed at NASA-Langley Research Center, to aid a user of the ADS computer program in design optimization of structures. Because ADS has three levels of options (strategies, optimizers, and one-dimensional searches), the user has approximately 100 combinations from which to choose. The expert system aids the user in choosing the best combination of options for solving a particular problem using ADS. The system is written in LISP, contains about 200 rules, and executes on DEC-VAX and IBM PC/AT computers.

2.2.2.5 EXPERTISE

This is an expert system, developed by Structural Analysis Technologies, Inc., for assisting civil/structural engineers with earthquake-resistant design of building structures. A design engineer, designing buildings on a routine basis, does not have knowledge in all facets of the earthquake-resistant building design process. Different experts are needed to provide consultation and inputs to the design engineer on various different aspects of the earthquake-resistant design of buildings, e.g., Geologists, Seismologists, Geotechnical Engineer, Structural Analyst, Structural Designer, Structural Dynamist, Statistician, as well as experts from legal, financial, regulatory and public safety related fields. The system duplicates these diverse inputs and expertise. The expert system is developed in "C" language using our own in-house shell, and is operable on PC's. The program is menu-driven, utilizes color graphics, and is linked to several databases. The architecture for EXPERTISE is shown in Figure 2-3.

2.2.2.6 GARI

GARI is an expert system developed by researchers at the Laboratoire de Marcoussis and the National Polytechnic Institute in France. Given a description of a part to be machined in terms of features (e.g., trapped holes, bores, grooves, notes, etc.), GARI attempts to determine a plan detailing the cuts to be executed, their ordering, the surfaces by which the part is to be clamped, the tools to be used, etc. Example rules in the knowledge base include

"if a hole, H1, opens into an other hole, H2: then machine H2 before H1 (avoiding risk of damaging drill)"

"if a finishing cut is executed on a grinding machine and another cut is executed on a different machine, then do the latter first."

By introducing a weighting metric on its various rules, GARI is capable of dealing with contradictory advice which can occur during processing. Faced with conflicting strategies (e.g., drill hole1 before hole2 and drill hole2 before hole1), GARI can tentatively retract the piece of advice with the lowest weight.

2.3 INTELLIGENT DBMS FRONT-ENDS

Most of the research in the joint application of artificial intelligence and database management techniques has concentrated on knowledge-based systems that act as database assistants in the areas of query optimization, data access through natural language interfaces, and deductive databases. Natural language DBMS front-ends like RESADA (Ref. 53) and CO-OP (Ref. 54) have the capability to parse simple queries expressed in a restricted set of conversational English into database access requests. Deductive databases (Refs. 55-57) use mathematical logic to represent knowledge about the database domain, deducing new facts from the data available in the database.

Since the focus of the research described in this report is on knowledge-based systems that act as database users rather than database assistants, most of the work involving intelligent database front-ends need not be further discussed. However, two particular prototype systems are worth examining for their flexible approach to interfacing the intelligent front-end with the DBMS.

2.3.1 DADM

DADM (Deductively Augmented Data Management) (Refs. 58-62) is a combination of a file of general knowledge with an associated reasoning engine and a file of specific knowledge with an associated searching engine. The general knowledge consists of a set of domain-specific assertions expressed in first order predicate calculus, and the file of specific knowledge is supported as multiple databases by a single relational DBMS. A unique feature of DADM is that, in recent versions, the system has been implemented on heterogeneous hardware components: A reasoning engine running on a LISP machine, a relational database supported by a specialized database machine, and a query/reply translator running on a DEC VAX 11/780.

The reasoning engine functions primarily as an intelligent interface to the database. The reasoning engine uses knowledge about the problem domain to

construct database search strategies to locate answers to queries from the user. The corresponding database queries are sent to the VAX for conversion to the database syntax before being passed onto the database machine. Replies from the database to the reasoning engine follow the reverse path.

One of the first applications of DADM is a "Manager's Assistant" (Ref. 63) to aid in corporate project monitoring and planning. The knowledge base for this application consists of general managerial expertise on such topics as staffing, plan-versus-actual discrepancies, personnel turnover, etc. The databases for the Manager's Assistant include a manager-specific database of planned charges to projects and a general database of actual charges to projects.

2.3.2 FRED

FRED (Front-end for Databases) (Ref. 64) is a portable, natural language database interface for use with multiple databases. FRED uses a semantic case grammar to translate restricted natural language queries into an internal set of case frames. FRED's query planner and database processor transform the case frames into a set of database queries. FRED's query planner and database processor are shown graphically in Figure 2-4.

The query planner converts the case frames that are generated by the natural language parser into a query expressed in V/DELPHI (Virtual Database Enquiry Language for Portable Heterogeneous Interfaces) using its planning rules. The planning rules contain the knowledge required to transform the network of case frames into the linear string of tokens that represents the equivalent query in the nonprocedural V/DELPHI. At this point, the query is based on a universal relation; i.e., all the data required is imaged to be available in a single virtual database table. The next step is the V converter, which uses domain-to-data mappings to translate the requested data items into the set of those available in the databases. The V converter applies the universal relation definitions that represent the mappings necessary to combine all the actual database tables into the virtual universal relation. The TROLL transformer uses DBMS-specific query language transformation rules to develop queries for the individual databases.

In its current implementation, FRED is linked to an Oracle database supporting a database of medical information. The query planner is capable of handling queries of "low to medium complexity (Ref. 64)". FRED's database processor has rules for generating queries in three languages: SQL, FOCUS, and Dataquery.

2.4 COUPLED EXPERT SYSTEMS AND DATABASES

This section focuses on expert systems that act as database users. In the prototype systems described below, the emphasis is on the database as a source of information for the expert system.

2.4.1 RX

RX (Refs. 65-67) is a knowledge-based medical research system that formulates causal hypotheses and statistically tests them for validity against a clinical database of medical information. RX consists of a discovery module, a study module, a knowledge base, a time-oriented database, and a statistical analysis package. The discovery module generates hypotheses of the form "A causes B". Using information from the knowledge base, the study module transforms these hypotheses (or hypotheses input by the user) into specialized models to test statistically against the data in the database. The knowledge base is organized into a set of frames which describe existing medical information (currently limited to facts about systemic lupus erythematosus), statistics, and RX system knowledge.

The time-oriented database allows sequential sets of values to be specified for the attributes which describe a patient's condition. The data available to the system at run-time is a small slice of a very large clinical database, containing 50 patient records selected by the programmer. The run-time database is managed by an ad hoc data manager with a customized interface to the rest of the system.

2.4.2 Expert Orthodontics Tutor

The expert tutor in orthodontics described by Kanamori (Refs. 68-69) combines an expert system for comparative case study analysis with a relational database that maintains detailed orthodontics case studies and a image base of skill line drawings for graphics display. The motivation behind the system is to allow the student the opportunity to compare treatment results and to see the treatment's progress over an accelerated time scale. In a typical interaction with the system, the student describes a case along with a proposed treatment plan, and the system displays the results of similar cases, allowing the student to infer the validity of the diagnosis and proposed treatment plan. The rationale for using this approach of comparative analysis is that a firm diagnostic logic for orthodontics treatment does not exist.

The associated DBMS was specially constructed for the project; it has a data manipulation language that is based on the INGRES DBMS query language, QUEL. The database contains thirteen relations describing diagnoses, treatments, skill features, etc. The expert system generates detailed queries

to the DBMS for case studies that match (within tolerances) the parameters of the case in question. The auxiliary image base is a file of x- and y-coordinates for the digitized skill line drawing accessed and displayed by a specialized FORTRAN program that is linked with the tutor.

2.4.3 HICOST

HICOST (Refs. 70-71) is used to develop preliminary cost estimates for high-rise buildings. It is used by HI-RISE (Ref. 70) to evaluate competing preliminary designs based on structure cost. Given the topology and geometry of a building and HI-RISE's preliminary choice for the structural system and elements of the superstructure, HICOST produces an estimate of the building's cost.

HICOST is implemented as a rule-based expert system with algorithmic computational functions, which queries a database for specific data on component costs. It is a tightly coupled integration of several expert system tools and a relational database. The cost estimator is a hierarchically-organized production system with rules written in PSRL (Production Schema Representation Language) (Ref. 72). Data objects are defined in frames in SRL (Schema Representation Language) (Ref. 73), and cost data is stored in a relational database supported by INGRES (Ref. 74). The production system computes the estimated cost from aggregate subsystem costs. Access to the database is through demons attached to HI-RISE cost frames. Each subsystem data element has an associated cost value. Accessing the cost value of a basic component (beam, column, etc.) triggers a demon which invokes a function (written in Franz LISP), which in-turn calls a procedure written in "C". The "C" procedure uses the INGRES procedural query language (EQUEL) to access the database and return the item cost to the estimator.

2.4.4 Insurance Expert System

An expert system for customizing insurance policies (Refs. 75-78) is under development at New York University in cooperation with IBM. The system has a knowledge base that includes formal rules (from actuarial science, finance, and insurance law), and informal heuristic rules (for generation of policies, typical customer requirements, identifying legal and corporate constraints) tightly coupled with a relational database containing customer data, tables for mortality, interest, etc. The expert system can access a library of mathematical subroutines to perform extensive computations. The knowledge base and inference engine are implemented in PROLOG, and the database is managed by IBM's SQL DBMS. Database queries are translated from PROLOG to DCBL (a variable-free subset of PROLOG) and from DCBL to SQL.

A small demonstration prototype has been implemented, without interfaces to an actual DBMS or to a mathematical subroutine library. The implementation environment consists of an IBM 4341 running VM/CMS.

2.4.5 SICAD

SICAD (Standards Interfaces in CAD) (Ref. 79) is a knowledge-based approach to standards processing. SICAD uses a custom knowledge base and inference mechanism to access standards represented in a knowledge-based format. The knowledge base contains three types of knowledge about standards:

- Classifier Trees are used to relate engineering terminology to provisions of a standard.
- Information Network is a network of decision tables that represent the provisions of a standard. The decision tables are represented as FORTRAN code.
- Mappings are used to relate data items in a standard to data items in a design database.

SICAD is intended to be interfaced with a design program that needs to perform compliance checking. The design program invokes SICAD to identify and check applicable provisions. The checking process uses a goal driven search strategy to determine all data items required to evaluate a provision. SICAD automatically accesses this data in the program's database through the mappings stored in the knowledge base. The structure of SICAD is shown in Figure 2-5.

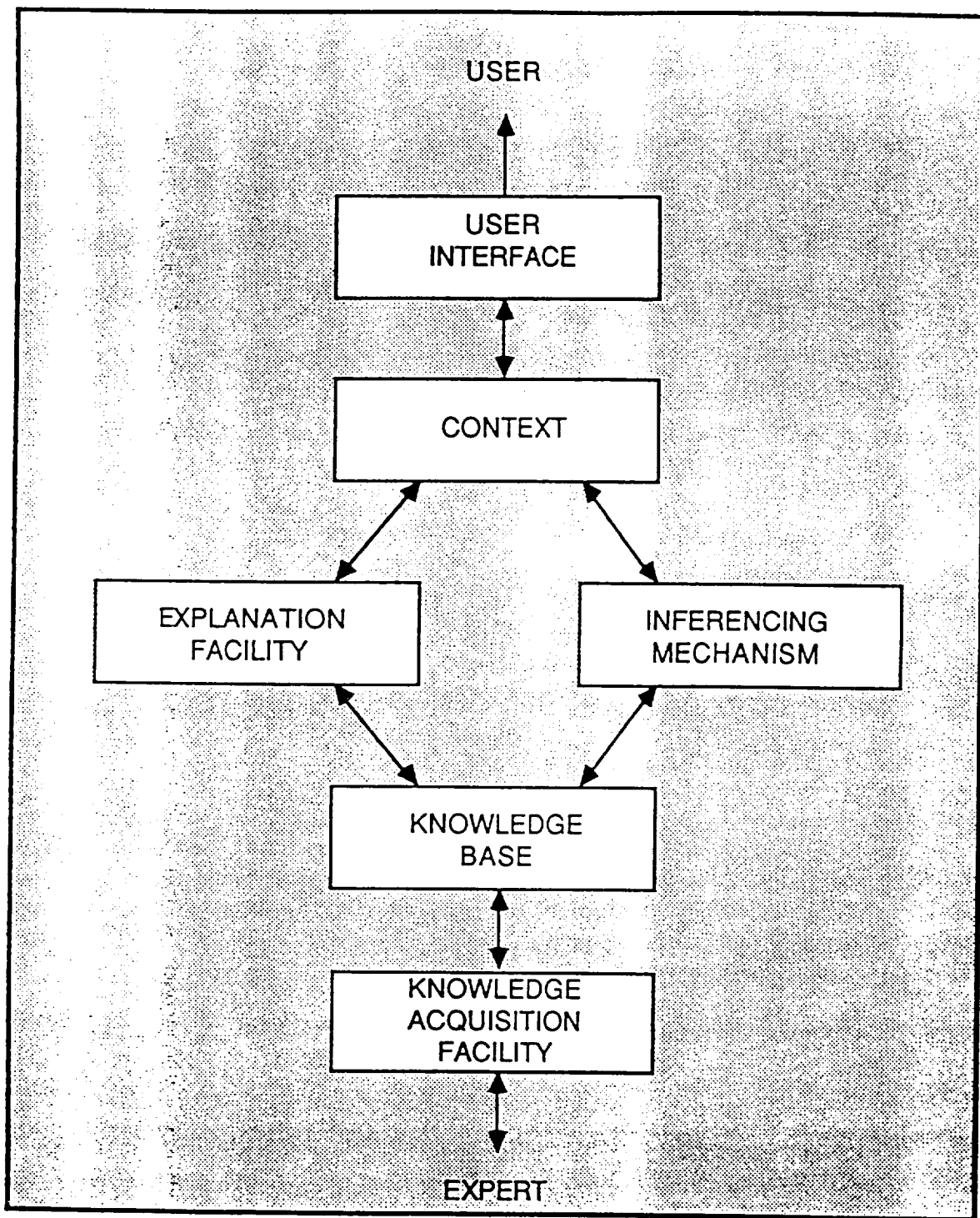


FIGURE 2-1: SCHEMATIC VIEW OF A TYPICAL KNOWLEDGE-BASED EXPERT SYSTEM

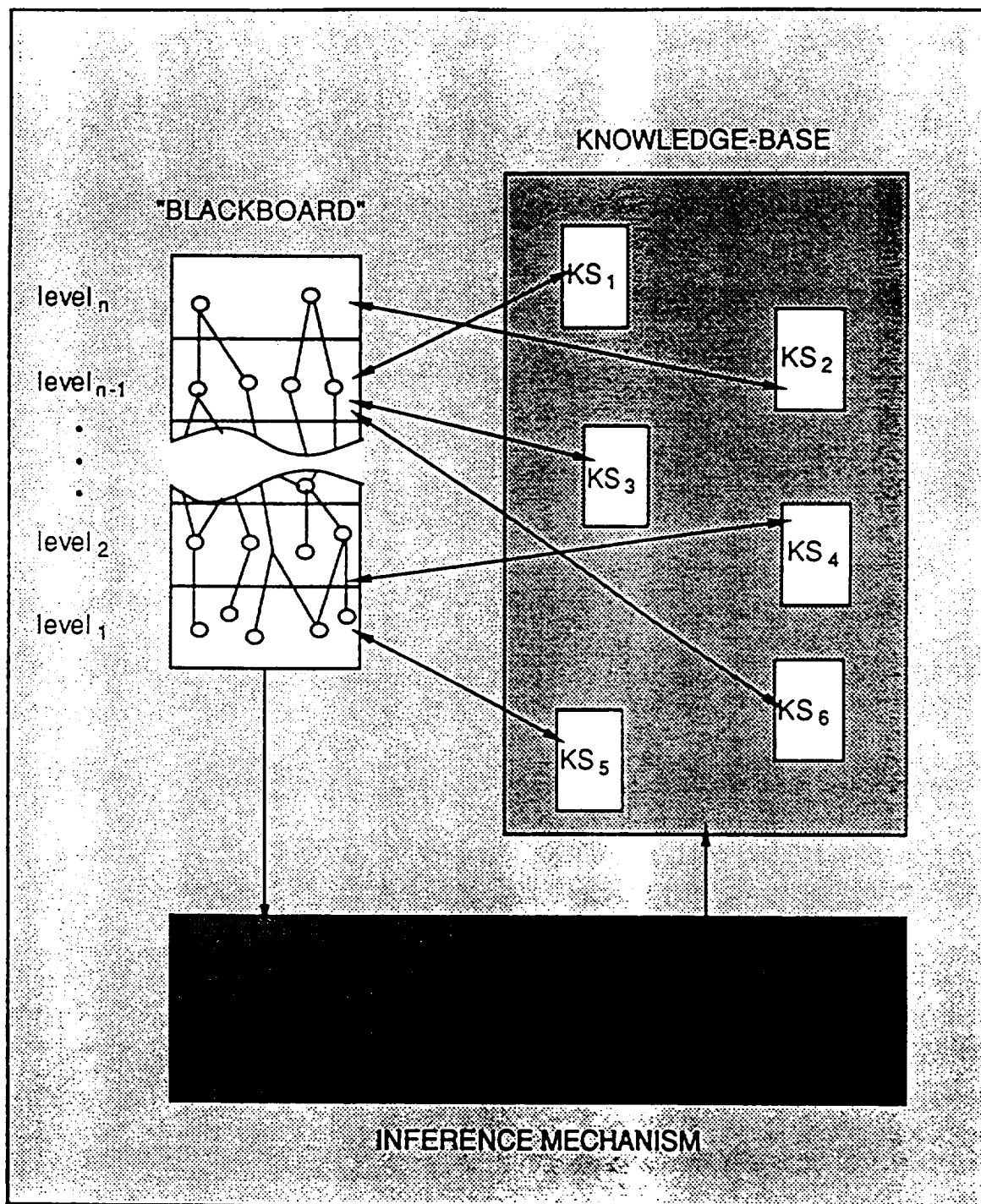


FIGURE 2-2: THE "BLACKBOARD" ARCHITECTURE

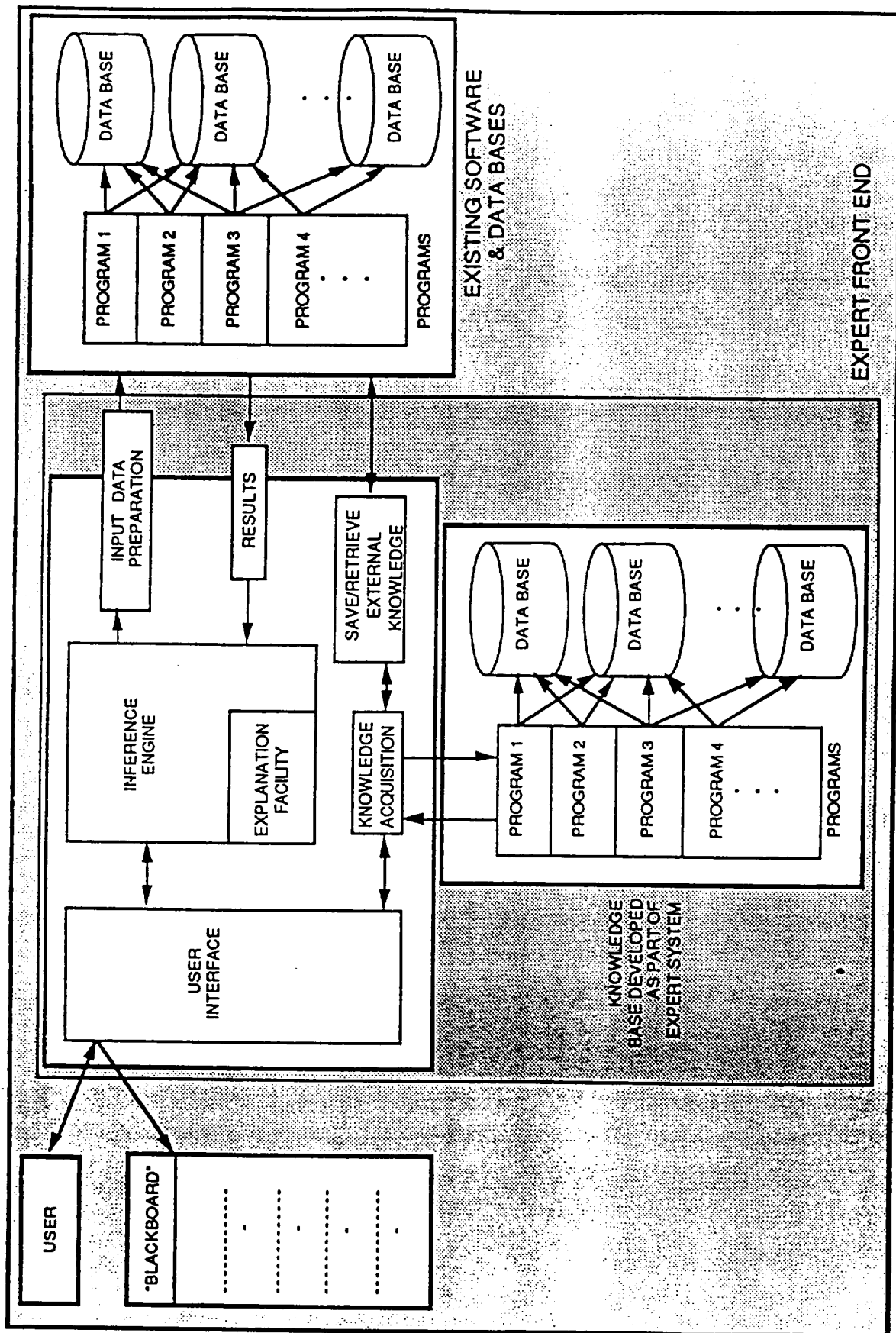


FIGURE 2-3: THE ARCHITECTURE FOR EXPERTISE

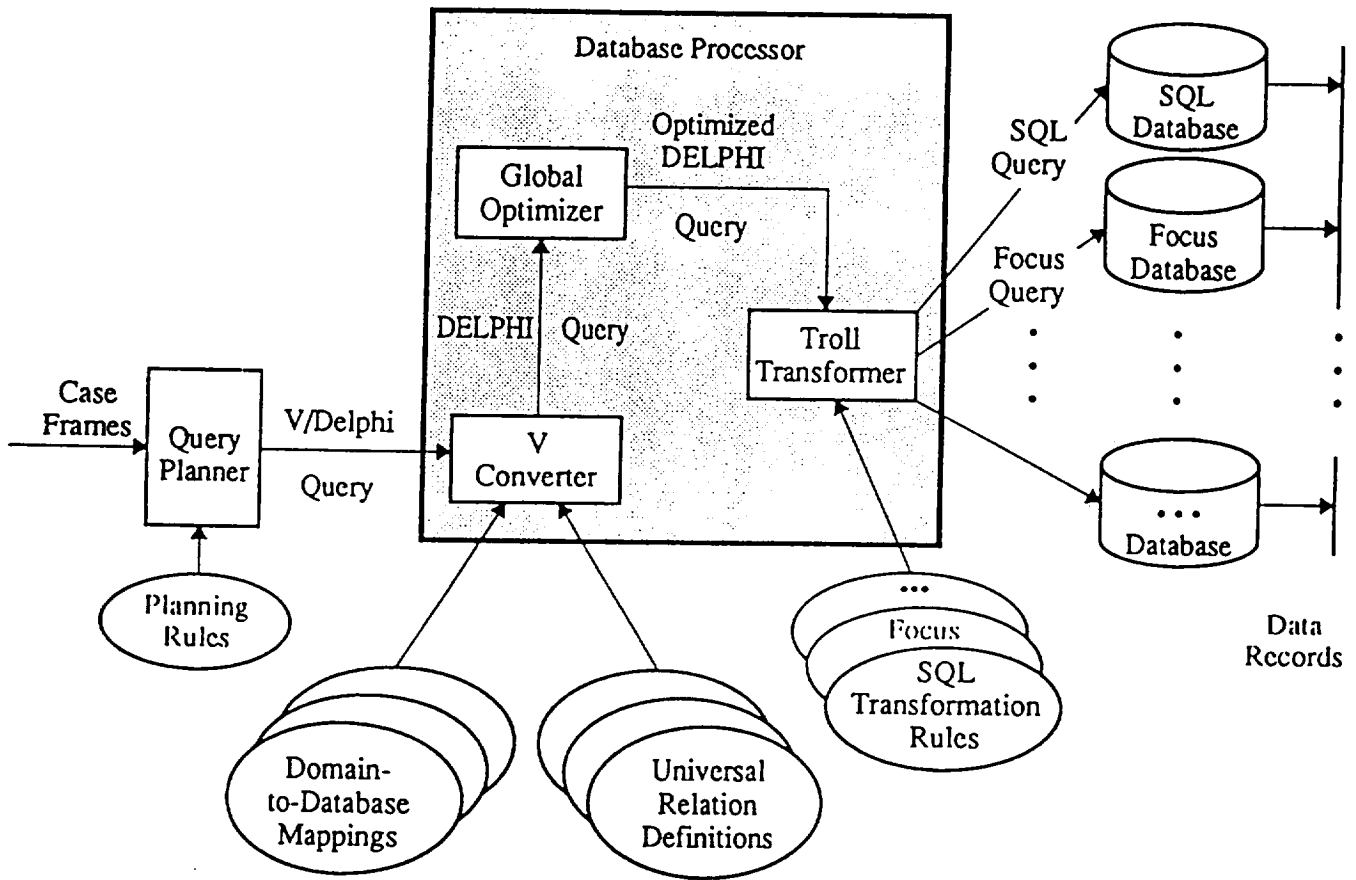


FIGURE 2-4: QUERY PLANNER AND DATABASE PROCESSOR FOR FRED

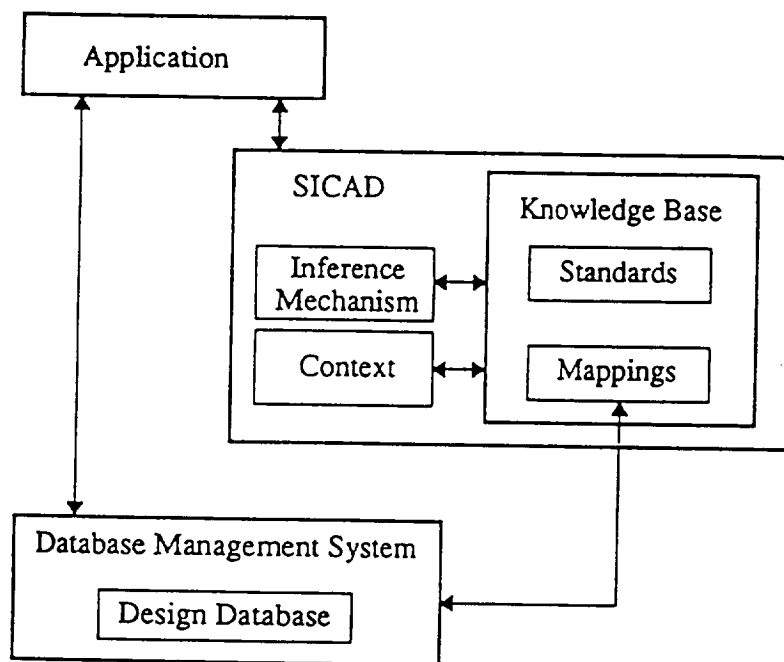


FIGURE 2-5: SICAD SYSTEM STRUCTURE

3. FUNCTIONAL SPECIFICATIONS AND ARCHITECTURE FOR EXPERT SOFTWARE FOR STRUCTURAL ANALYSIS AND DESIGN OPTIMIZATION OF AEROSPACE STRUCTURES

In this chapter, the functional specifications and architecture of the expert software, integrated expert package for the structural design/analysis/optimization of aerospace structures, are discussed. First, however, the overall aerospace design process and the integrated analysis and design optimization technology are reviewed.

3.1 AEROSPACE DESIGN PROCESS

The design process for aerospace structures involves many analysis/design iterations, exchanges of large amount of data and multiple interaction on decisions among a variety of technical disciplines. Typically, the design process goes through several stages, ranging from early conceptual design to the detailed analysis of a final design. For each stage and cycle of the analysis and design, a large number of parameters are investigated, and numerous algorithmic computer programs are employed. During these design cycles, the engineer also utilizes a large amount of data, either via computerized databases or as past project reports and experiential knowledge.

The various activity levels on a standard engineering project in the aerospace industry are shown in Figure 3-1. The main activities associated with the analysis and design part of the complete process can be divided into the following phases.

1. Conceptual Design
2. Preliminary Design
3. Structural Analysis
4. Structural Design and Optimization
5. Development of Fabrication Specs and Drawings

This study is primarily limited to the structural design process starting from preliminary design up to final design through the iterative cycles of structural analysis and design optimization. It is assumed herein that conceptual design has already been performed and the criteria and loadings have already been developed.

In order to develop the expert software for integrated structural analysis and design optimization, it was therefore important to understand the basic steps

of this process and identify the nature and extent of the knowledge/data and decision making inferences needed in the process. This is discussed below in more detail.

3.2 INTEGRATED ANALYSIS AND DESIGN OPTIMIZATION

The analysis and design optimization process is iterative in nature. The basic steps of this process are the following:

1. Finite Element Stress Analysis
2. Design Optimization, Including Sensitivity Analysis

These steps are discussed below in more detail, along with major decisions associated with these steps and the critical questions asked at each step:

3.2.1 Finite Element Stress Analysis (Ref. 80)

The main objective of finite element analysis is to determine the behavior of the structural design being analyzed. This may include analysis of global displacements under static loads, global accelerations under dynamic loads, local stresses, fracture or fatigue behavior, stability evaluation, or buckling or post-buckling response. Thus, depending on the objective of the analysis, the analysis types may be one or more of the following, among others:

- Global Static Analysis
- Global Dynamic Analysis
- Detailed Local Stress Analysis
- Heat Transfer and Thermal analysis
- Flutter Analysis
- Fatigue or Fracture Analysis
- Stability Analysis
- Buckling or Post-buckling Analysis
- Acoustic Response Analysis
- Impact Analysis
- Nonlinear Stress Analysis
- Damage Tolerance Analysis
- Others

The finite element analysis involves the following major tasks:

3.2.1.1 Development of Specifications for Finite Element Analysis

This is an important step, before embarking upon the actual analysis, and involves the following subtasks:

- Analysis Planning
- Definition of the Physical Structural Design
- Definition of the Structural Context
- Definition of the Purpose of Analysis
- Formulation of the Problem
- Definition of the Solution Context
- Definition of the Solution Requirements and Presentation

A lot of judgement, reasoning, and major decisions are involved in these subtasks. Some examples of the decisions involve answering the following types of questions:

- Who is to carry out the analyses? What are the available resources?
- Is the person or the team adequately experienced or qualified to perform the analyses?
- Which type of computer would be used for such analyses, a mainframe, work station or PC?
- What software would be used for such analyses?
- What data is required for such analyses?
- What part or component of the structure (e.g., an aircraft) will be analyzed (e.g., a wing, fuselage, etc.)? What will be the effect of the adjoining component on this structure?
- What is the purpose of the analysis? Initial Design?, Detailed Stress Analysis of the Final Design? Local Stress Analysis? Stability Analysis? Flutter Analysis? Failure Analysis?, etc.
- Does the proposed analysis require detailed information about stresses or distortions in the immediate vicinity of a physical feature (e.g., a notch or hole)?
- Is the physical feature likely to have any noticeable effect on global distribution of stresses?
- What are the requirements for output and presentation of results?
- How the real problem should be formulated in finite element terms? What would be the expected structural behavior which needs to be investigated by finite element solution?

3.2.1.2 Modeling for Finite Element Analysis

Modeling of structures for finite element analysis is the activity whereby a structural design problem is formulated in terms suitable for solution using a finite element computer program. A modeling strategy is needed at the outset. It is usually better to start from the fundamental position that the aerospace structures and components are so complex that the lowest level of significant structural detail (e.g., small holes, grooves, cracks, etc.) can not be explicitly represented by a single comprehensive finite element model, and

multilevel analyses (global to local) are required, preferably (but not necessarily) using sub-structuring.

At the global level, the predominant issue is the extent to which local features should be represented explicitly. At the local level, major issues include interfaces with adjoining structures. At both levels, the primary issues include the following:

- Mesh Refinement
- Load Application
- Material Models and Representations
- Inertial (Mass) Representation (for dynamic problems)
- Symmetry
- Boundary Conditions
- Types of Elements
- Others

Major decisions are required to ensure that the finite element model can be developed and analyzed within the available cost and time restraints, while at the same time providing the desired accuracy and reliability of results. Many of the decisions made are purely heuristic in nature, i.e., based on judgement and experience.

The primary issues to be considered in the development of a finite element model, the associated major decisions and the questions asked are discussed below:

Many aerospace structures are designed and built as roughly rectangular assemblies of skins, spars, ribs, and stiffeners. The pattern of actual members may, many times, dictate a natural mesh, by following intersections and then improving proportions by subdivisions of slender panels.

In areas of abrupt geometric changes or load concentrations, a finer (preferably graded) mesh may be used. Stress gradients, principal stress contours, acceptable accuracy and errors, may also define mesh refinement; as well as cost, time and convenience (as discussed above). It may also be desirable to utilize substructuring to break down the analysis into manageable parts, especially if there are several identical (or repeated) substructures.

The mesh refinement is also affected by the "aspect ratio" of the finite elements used, the depth-ratio, as well as symmetry-factor.

The objective of the analysis is also a major factor in deciding the mesh size, e.g. a coarse mesh can be used if the objective is initial sizing of structure,

global optimization, global dynamic response or dynamic stability analysis. A fine mesh is required for stress analysis of final design, fracture mechanics, local response, or acoustic response.

For local modeling of structural features, either explicit or implicit modeling can be carried out. In explicit modeling, the important geometry features are represented by appropriate local mesh and element selection and blended with the basic mesh. In implicit modeling, the basic mesh is continued with no more than minor changes, but the section properties and/or element properties of the elements are modified to simulate the effect of the features. Thus, major decisions are required in regard to mesh refinements. Some examples of the decisions, and the associated questions, include the following:

- Is accuracy the principal criterion for mesh refinement?
- Are cost, time, and convenience the main criteria for mesh refinement?
- What is the purpose of analysis?
- What are the available resources?
- What hardware and software will be used?
- Is it appropriate to use a "natural mesh," defined by structural geometry and intersection of components?
- Are there any abrupt changes in geometry or loading?
- Are there any concentrated loads being applied?
- Are there any holes and opening, etc?
- What type of finite elements will be used, and what are their aspect ratios?
- What kind of structural behavior is expected, given the structure geometry and loadings?
- Are there any structurally significant features or fixtures? If so, should they be modeled explicitly or implicitly (e.g., by using "equivalent" elements)?

3.2.1.3 Substructuring

Substructuring is an approach for breaking down a finite element analysis into manageable parts. It provides significant savings in solution cost and time, especially if there are several identical and repeated substructures. Examples of major decisions and the associated questions, with regard to substructuring, are the following:

- Are there significant changes in the relative geometry of adjoining structures?
- Are different analysis teams assigned to different tasks?
- What are the hardware and software available?
- Are localized iterations required, for example for local nonlinear or contact problems?

- Are several components very sparsely connected?
- Are there multiple, repeating substructures?

3.2.1.4 Load and Inertia Modeling

A. Load Modeling

The preparation of loading data, e.g., for aerodynamic loadings, which depend on element formulation as well as other factors (wholly extraneous to the analysis), can sometimes become the dominant task in analysis. This may be due to the fact that aerodynamic data are usually derived as load parameters from a different mesh of the structure, and also the fact that representation of continuous loading is only possible within the limited capabilities of the chosen structural elements. Examples of major decisions and the associated questions in load application may include the following:

- How should the given loadings be converted into nodal data for the finite element mesh?
- What kind of loading combinations be used to get maximas of various response quantities? What is the purpose of the analysis?
- How important is the local response?
- What are the capabilities of the software?

B. Inertial (Mass) Representation

If local accuracy and local natural modes are important, inertial modeling becomes a difficult task and requires a significant judgment in making decisions about how to distribute the masses.

Sample examples of decisions, required for inertial (mass) representation, include the following:

- What is the purpose of the analysis? Eigenvalue solution or detailed dynamic stress analysis?
- What kind of hardware and software are available?
- What is the expected behavior of the structure? Inclusion of how many natural modes will get the required accuracy?
- What is the refinement of the model mesh?
- Is it necessary to use a consistent-mass matrix, or a lumped mass approach is sufficient?
- At what nodes should the masses be lumped to obtained desired accuracy?

3.2.1.5 Selection of Elements

Selection of elements in a finite element analysis is a major subtask and determines, to a great extent, the accuracy and reliability of results. It is usually necessary to use a combination of elements for different components of the structure, depending on whether they are one-dimensional, two-dimensional, or three-dimensional. For one-dimensional components or structures, either a truss-bar or a beam element may be used, depending on whether axial behavior only or axial-flexural (and shear) behaviors are important. For two-dimensional components or structures, an axisymmetric, plane stress or strain, or plate elements may be used. For axisymmetric components or structures, with axisymmetric or non-axisymmetric loadings, an axisymmetric element may be used. If out-of-plane shear is not important, planar (plane stress or strain) elements may be used. However, if out-of-plane shears are important, plate element must be used.

For three-dimensional structures, the usual choice is between plate/shell and 3-D solid elements. For structures such as an aircraft fuselage or an undercarriage mounting bracket, the decision is obvious. But, the decision is not that simple for structures such as solid missile wing, the undercarriage oleo leg, the thick skin at the root of a composite wing, the one-piece forged or machined airbrake. For such structures, a balanced judgement is needed, weighing the analysis purposes and accuracy on the one hand against cost and complexity on the other.

Other choices include considerations of lower-order element (e.g., linear isoparametric) versus higher order elements. Lower-order elements are preferable for modeling structural features and details since they are easier to use. Higher-order elements are preferable for continuum analysis because of their better accuracy and economy in regions of higher stress concentration. Stress-based or hybrid elements are preferable for boundary representations which are closer to stress requirements. The other choices are between the use of quadrilateral or triangular elements. Generally, quadrilateral elements out-perform triangular elements but triangles are often easier to fit into graded meshes.

Sample examples of decisions, and associated questions regarding selection of finite elements, include the following:

- What is the purpose of analysis? Detailed stress determination?
Displacement determination?
- What kind of software or hardware is to be used?
- What accuracy is desired? What are cost considerations?
- Is the structure or component primarily 1-D, 2-D or 3-D?

- Is the structure of characteristic shell or plate form or built up from such components?
- Is thickness small compared with other significant dimensions?
- Are through-thickness or out-of-plane shear stresses likely to be significant?
- Is stress distribution through the thickness important?

3.2.1.6 Symmetry

With the use of symmetry, and the appropriate constraints, the finite element representation can be significantly simplified. For reflective or cyclic symmetries, the structure can be analyzed as single segments subject to appropriate loadings and kinematic constraints. In the extreme case of axial symmetry, a single cross-section rotated around the axis of symmetry can be analyzed. If there are N repetitions of the basic region, then, in general, the structural problem can be solved completely by performing N analyses of the basic region. If the loading cases are also symmetric, the number of solutions can be reduced to the number of symmetric loading conditions.

3.2.2 Structural Optimization

There are two major structural optimization procedures: (1) Structural Optimization based on Mathematical Programming (MP) techniques, and (2) Structural optimization based on Optimality Criteria (OC) techniques. These procedures, and the associated judgments/heuristics and decisions are discussed below:

3.2.2.1 Structural Optimization Procedures based on Mathematical Programming (MP) technique [Ref. 81].

The major components of Structural Optimization procedures based on MP techniques include the following:

- Problem Formulation
- Finite Element Analysis
- Sensitivity Analysis

3.2.2.1.1 Problem Formulation

The most dramatic advances in the area of structural optimization in the last ten years have come in the area of problem formulation. It is now recognized that simple coupling of a finite element analysis program, even including sensitivity calculations, is not adequate to provide the efficiency necessary for the design of practical structures. One of the principal advances in this regard have been the development of high quality approximations to structural

responses which can be used for optimization without the multitude of detailed analyses that was previously required. Coincident with this has been the formalization of such concepts as design variable linking and temporary constraint deletion to further improve design efficiency and realism.

Design variable linking has been used for some time as a means of imposing practical considerations as well as reducing the number of design variables that must be considered in the optimization phase. For example, several finite element dimensions can be controlled by a single design variable as a means of imposing structural symmetry. More importantly, it is now generally agreed that treating (for example) the thickness of each element in the analysis model of an aircraft wing as a design variable is not realistic from manufacturing considerations, and can even introduce significant errors into the analysis model itself. Thus, design variable linking has the practical usefulness of keeping the design process realistic and the theoretical usefulness of reducing the difficulty of the optimization task simply by reducing the design problem size.

The concept of constraint deletion is also nothing more than introducing realism into the automated design process. Experienced engineers seldom consider all design constraints simultaneously since some are easily identified as being noncritical. However, there has been a tendency to routinely include all stress, displacement, frequency, etc., constraints throughout the design process. The disadvantage of consideration of all constraints is that it is necessary to calculate the sensitivity of all constraints to the design variables. Therefore, it has been found that, because the current design stage is only a step toward the optimum, the logical approach is to delete from consideration all constraints that are not currently critical or potentially critical. For example, if a particular stress constraint is far from its limit, it can be ignored for a while and included later if it becomes near critical.

In addition to the concept of temporary constraint deletion, it is often most efficient to ignore some constraints early in the design process and include them later as the design is refined. Consider, for example, the case where stress, displacement, frequency and aeroelastic constraints must be considered in the design. It may be most reasonable to first perform one cycle of the classical fully-stressed design method, even though this design is likely to violate other constraints. Then, using this as a starting point, one or two design iterations can be performed with the displacement constraints included in addition to stress constraints. Following this, frequency constraints can be added, and finally the aeroelastic constraints can be added. The basic concept here is to first solve the easy problem to provide a good initial design for the more complex later ones. It is noteworthy that this is what is usually done when optimization is not used, simply because it is most efficient. By ignoring complex constraints early in design process, we save

considerable computational costs by virtue of the fact that they need not be evaluated. Furthermore, as new constraints are added, if they are not critical at the start of the new optimization, only one constraint evaluation is initially needed. The key idea when using this approach to automated design is that, as the complexity of the constraints considered is increased, all "lower level" constraints are also included. Thus, the optimum solution (assuming that it is unique) must be the same as that obtained by including all constraints from the start, but at a much lower cost.

While such concepts do much to improve the efficiency of the optimization process, one of the significant recent advancements has been the development of approximation concepts themselves. Here it has been recognized that, if the original problem can be approximated in some explicit form, then this approximation can be used for the actual optimization phase. A new approximation can then be created at the proposed design point and the process repeated until it has converged.

In structural optimization, it is often possible to make very high quality approximations to the response quantities, where the approximation are not linear, but are explicit and are often separable. This allows for solution of a more accurate approximate sub-problem and, where an explicit dual of the problem can be written, duality theory can be used for its efficient solution.

The key idea in approximation techniques is to create a high quality approximation to the original finite element based problem. The optimization is then performed on this approximate problem and a new proposed design is produced. The structure is then analyzed in detail and the process is repeated until it has converged to a satisfactory solution.

Some of the major decisions, and the questions associated with design optimization problem definition, are the following:

- Is the problem (and the model) consistent, bounded, and feasible?
- Are there any useful transformations? If so, what transformations?
- Is it possible to start with some approximations initially? If so, what approximations?
- Is it possible to simplify constraints initially, or use major constraints only, initially, and then add other constraints in the next iterations?
- What should be done about discrete variables?
- Are variables and constraints scaled?
- Are there redundant constraints? Possible degeneracies?
- What mathematical form should be used (e.g., Linear, Quadratic, Convex, Monotonic, etc.?)
- How many iterations may be required for problem solution?

3.2.2.1.2 Sensitivity Analysis

In the context of structural optimization, sensitivity analysis is taken to mean the rate of change of structural response with respect to the independent design variables. The structural response quantities include stress, strain, displacement, eigenvalue, flutter speed, buckling load, etc. The design variables may include member sizes, joint locations in a discrete structure, shape definition in a continuum structure, and material properties in a composite structure.

The sensitivity information provides a direction and amount by which the design variables must be changed in order to reduce weight or reduce some critical stress, as examples. In optimization, since many variable will be changed, this information can be used mathematically to find the best way to change them simultaneously. Most modern optimization algorithms require sensitivity information in order to efficiently direct the design process.

There are three standard methods for performing sensitivity analysis: (1) Direct Method, (2) Adjoint or Dummy Load Method, and (3) Material Derivative Method. These methods are briefly discussed below:

(1) Direct Method

Noting that the sensitivity of stresses can be directly calculated from the sensitivity of displacements, the sensitivity of the displacements can be calculated by implicitly differentiating the equation of equilibrium.

$$\mathbf{K}\mathbf{u} = \mathbf{P} \quad \text{Equation (3.1)}$$

where, \mathbf{P} is the vector of applied loads
 \mathbf{u} is the vector of displacements, and
 \mathbf{K} is the Structure Stiffness Matrix

The above equation can be rearranged, as follows:

$$\frac{\partial \mathbf{u}}{\partial x_i} = \mathbf{K}^{-1} \left[\frac{\partial \mathbf{P}}{\partial x_i} - \frac{\partial \mathbf{K}}{\partial x_i} \mathbf{u} \right] \quad \text{Equation (3.2)}$$

where it is noted that the rate-of-change of the structure stiffness matrix, \mathbf{K} , is the sum of the rates of change of element stiffness matrices with respect to the particular variable.

The sensitivity of the stress can then be recovered from

$$\frac{\partial \sigma}{\partial x_i} = \left[\frac{\partial}{\partial x_i} \underline{S} \right]^T \underline{u}_e + \underline{S}^T \frac{\partial \underline{u}_e}{\partial x_i} \quad \text{Equation (3.3)}$$

where \underline{S} is the stress-displacement relationship and \underline{u}_e is the vector of element nodal displacement.

In many cases, the necessary sensitivity of the element stiffness matrices can be calculated directly. for example, if the design variable is the cross-sectional area of a bar, the element stiffness matrix is the area time a constant (geometric and material) matrix, so that its derivative with respect to the member area is just this constant matrix. For higher order elements or geometric design variables, a finite difference method may be used. This is called a "Semi-Analytical" method.

(2) Adjoint or Dummy Load Method

In this method the constraint equation is differentiated as:

$$\frac{dg}{dx_i} = \frac{\partial g}{\partial x_i} + \underline{Z}^T \frac{\partial \underline{u}}{\partial x_i} \quad \text{Equation (3.4)}$$

where $\underline{Z}_j = \frac{\partial g}{\partial j}$

If the constraint, g , is a stress constrain, \underline{Z} is the stress-displacement relationship. Substituting Equation (3.3) into Equation (3.4), we have

$$\underline{Z}^T \frac{\partial \underline{u}}{\partial x_i} = \underline{Z}^T \underline{K}^{-1} \left[\frac{\partial \underline{P}}{\partial x_i} - \left(\frac{\partial \underline{K}}{\partial x_i} \right) \underline{u} \right] \quad \text{Equation (3.5)}$$

This can be evaluated by solving the matrix equation

$$\underline{K} \underline{Q} = \underline{Z} \quad \text{Equation (3.6)}$$

The vector \underline{Z} is often referred to as an Adjoint or Dummy Load.

(3) Material Derivative Method

This method is useful for such problems as shape optimization of continuum structures. Essentially, the required information is calculated from

$$\frac{\partial \psi}{\partial x_i} = \int_B \sigma(\underline{u}) \epsilon(\underline{\lambda}) \underline{n} \frac{\partial r}{\partial x_i} dB \quad \text{Equation (3.7)}$$

where ψ is the displacement function for which the derivative is needed, $\sigma(\underline{u})$ are the stresses evaluated from the analysis and $\epsilon(\underline{\lambda})$ are strains calculated from the analysis using adjoint loads.

The major decisions and questions associated with sensitivity analysis are the following:

- Is it necessary to perform sensitivity analysis?
- What sensitivity analysis method should be used?
- Should any approximations be made for sensitivity analysis?

3.2.2.1.3 Optimization Algorithms For Mathematical Programming Methods

There are numerous Mathematical Programming algorithms for solving the optimization problem, but there is no clear consensus. The reason for this is that the choice of "search direction," that determines how the design variables are to be changed is not unique. Also, the method chosen is based to a large extent on the designer's philosophy. For example, a conservative designer may choose an interior penalty method since it produces a sequence of improving feasible (acceptable) designs. An unconservative designer, on the other hand, may select Sequential Linear Programming which produces a sequence of improving infeasible (unacceptable) designs.

In the 1960's, sequential unconstrained minimization techniques (SUMT) were popular. These methods used well developed unconstrained minimization algorithms and included constraints via some form of penalty which accounted for constraints becoming near critical or violated. Also, sequential linear programming methods were popular. These methods first linearized the objective and constraint functions and solved the resulting approximate problem by well known linear programming methods. In each case, the nonlinear design problem was solved by converting it to a form suitable for solution by well known methods.

In the 1970's, methods such as feasible direction methods, improved SUMT, the augmented Lagrange Multiplier Method (a form of SUMT) and reduced

gradient methods represented an improvement in both efficiency and reliability of the programming for nonlinear constrained optimization.

Also, during the 1980's, the earlier methods such as sequential linear programming method, the method of feasible directions, sequential unconstrained minimization, and others have continued to mature so that numerous algorithms are still available for the general nonlinear constrained optimization problem, and this maturing process can be expected to continue. The key point here is that optimization theory itself is fluid and, as the technology advances, the algorithms become increasingly efficient and reliable.

In the last decade, nonlinear programming methods have significantly matured. Many new software packages, based on nonlinear programming methods, have been developed. Some examples are given below:

Methods of feasible directions	CONMIN (Ref. 82)
Newton's Method with SUMT	NEWSUMT (Ref. 83) NEWSUMT A (Ref. 84)
Generalized Reduced Gradient Method	GRG2 (Ref. 85)
Generalized Reduced Gradient and other Methods	OPT (Ref. 86)
Sequential Quadratic Programming Methods and other Methods	IDESIGN (Ref. 87)
Variety of Methods	MICRODOT, ADS (Ref. 88)

The major decisions, and associated questions, regarding Structural Optimization based on Mathematical Programming (MP) methods, may include the following:

- What MP method should be selected?
- What submethod (e.g., line search, unconstrained, etc.) within the overall method should be selected?
- How should the user-defined program parameters (e.g., step size accuracy, penalties, termination, etc.) be selected?
- What criteria for accepting a solution should be used?
- Is it a global or a local solution?
- What are the active constraints?
- What should be done if solution errors are encountered?

3.2.2.2 Structural Optimization Based on Optimality Criteria (OC) Methods (Refs. 90-92)

Optimality criteria methods have been extensively studied and evaluated in the literature (Refs. 89-92), as already discussed. These methods are called "Indirect Methods" since the objective is to obtain a design that satisfies a certain specified criterion, and, by doing so, indirectly minimizes the weight of the structure. The criteria may be intuitive or derived mathematically, based on the nature of the problem. A standard method used for deriving the criteria is based primarily on differentiating the Lagrangian with respect to the design variables. They are evaluated by solving a linear system of equations that is obtained using derivatives of constraints with respect to the design variables and constraint function values. The optimality criteria are usually iterative in nature due to the essential nonlinearity of the constraints and the statical indeterminacy of the system.

In deriving the optimality criterion and developing the algorithms, full use is made of the knowledge of the behavior of the constraints imposed on the structure.

The original motivation for "optimality criteria" was to provide an approach for displacement constraints that is as simple as the stress ratio algorithm, and can augment it in practical automated sizing methods without having to resort to costly direct numerical search procedures. For the automated sizing problems, indirect minimization through satisfaction of an optimality criterion was found to be a simpler numerical task than direct minimization through a direct numerical search procedure. The reason was that the optimality criteria contain valuable gradient related information as a result of their derivation and take full advantage of the special structural properties of the problem. Direct numerical search methods have to numerically develop similar information while examining many points in the design space, and are unnecessarily general for a very special problem.

The constraints imposed on the structure may include the maximum allowable stress in each element, the displacement limits at one or more locations, system stability, dynamic stiffness, local element buckling etc. In addition to these, there may be limitations on the minimum and maximum sizes of the elements. An optimality criterion can be derived that includes all these constraints, and it may be desirable to find a design that satisfies this criterion. However, to develop an efficient algorithm based on such a criterion and effectively handle all types of constraints would be impractical and generally unnecessary. For most problems, it is more difficult to develop the algorithm than to derive the optimality criterion. In the case of most structures it is likely that one can predict the type of constraint which will be

the most active at the optimum and use the algorithm based on that constraint. Then, one can treat all other constraints as passive constraints. It is highly unlikely that all types of constraints will be active at the optimum. Sometimes, this point of view may not be correct and will not give an absolute minimum weight design. However, it gives a near minimum weight design, and the corresponding optimization algorithm will be efficient and easy to use for a structure with a large number of design variables. Even with this approximation to the overall problem, when the total number of constraints of the same type are large, it is advantageous to make additional approximations in order to reduce the computational effort.

The optimality criterion derived for all the constraints imposed on the structure is equivalent to the Kuhn-Tucker conditions of nonlinear mathematical programming. However, in deriving the optimality criterion and the corresponding structural optimization algorithm, some of the constraints are treated as side constraints in order to simplify the algorithm. A good example of this is the minimum and maximum size limits on the design variables. These constraints generally are not included in the constraint equations and do not enter the optimality criterion. The optimality criterion derived for structural optimization for a particular type of constraint gives information on the distribution of energy in the structure necessary to have a minimum weight design. The nature of energy depends upon the type of constraint.

In using Optimality Criteria methods, the analysis of the discretized structure is usually performed by the finite element method, similar to the Mathematical Programming methods. The redistribution of the material is then carried out by using a recurrence relation. The recurrence relation modifies the design variables, so that, in the design space, the initial design is moved towards a design that satisfies the optimality criterion. The recurrence relation contains two sets of unknown terms. The first set is related to the gradient of the constraints and the second set is related to the Lagrange multipliers. It is necessary to determine these unknowns before the recurrence relation can be used.

The efficiency and the convergence behavior of the algorithm depend on: (1) the recurrence relation used to modify the design variables; (2) the nature of the approximations made to derive the mathematical expression for the unknowns in the recurrence relations; and (3) how these unknowns are determined.

The major decisions and questions, associated with the use of Optimality Criteria Methods, may include the following:

- What optimality criteria will be used?
- How will the optimality criteria be derived?
- What active and passive constraints will be used?
- What approach will be used for the determination of Lagrange multipliers, viz., solution of linear equations or use of recurrence relations?
- What recurrence relations will be used to modify the design variables?
- What approximations will be made to derive the mathematical expressions for the unknowns in the recurrence relations?

3.4 FUNCTIONAL SPECIFICATIONS

Based on the review of the design process, as discussed in the previous sections, the functional specifications were developed for the expert software.

The expert software consists of an integrated environment of linked engineering tools available in an interactive fashion to the user for the design process of aerospace structures.

The main engineering activities covered include the following:

- Preliminary Design
- Finite Element Analysis
- Structural Optimization
- User Interface and Graphics/CAD Interface

Within each of these activities, a number of functions are available including administrative functions, procedural functions, and expert functions. These functions allow the user to perform domain-specific engineering; and assist him with key decisions, judgments and constraints-checking using an integrated knowledge environment.

The specific capabilities of the software are discussed below:

A. Preliminary Design

This function assists the user in performing a preliminary design of the proposed aerospace structure, while meeting key criteria and constraints. It is assumed, herein, that the conceptual design has been completed, and a general layout is available. The designer has to review the layout from a structural point of view, modify it as necessary, select preliminary member sizes, and assemble them to develop a preliminary design which can be used as a starting point in the iterative structural analysis/design optimization process to be able to obtain a final design.

B. Finite Element Analysis

This function assists the user in performing the finite element analysis of the structure or its components. The capabilities include the following:

- Development of the Finite Element Model
 - Selection of the Type of Finite Element
 - Development of the Finite Element Mesh
 - Application of Loadings and Boundary Conditions
- Analysis Execution
 - Static Analysis
 - Heat Transfer/Thermal stress analyses
 - Eigenvalue Solution
 - Dynamic Analysis
- Interpretation of Results

C. Structural Optimization

This function assists the user in performing optimization of structural design, including its components. The capabilities include the following:

- Sensitivity Analysis
- Problem Definition and Formulation
- Selection of Objective Function and Constraints
- Selection and Execution of Main Optimization Procedure
 - Mathematical Programming Method (Linear, Nonlinear, etc.) and Execution
 - Optimality Criteria Method and Execution
- Selection and Execution of Sub-Methods for Mathematical Programming Method
- Pre- and Post-Processing and Display of Design
- Interpretation of Results
 - Understanding of Design
 - Use in Next analysis or Design Iteration
- Selection of Member (Element) Sizes

D. User Interface and Graphics

This function assists the user with user interfaces, graphics, preliminary design, pre-processing and model development for finite element analysis, post-processing of finite element results, preprocessing for structural optimization, post-processing for structural optimization, as well as throughout the design process.

The capabilities for this sub-package includes the following:

- Interactive Graphical Interface
- Menu-Driven Interaction
- High Resolution Color Graphics
- Iconic Interfaces
- Random Access

3.3 ARCHITECTURE OF THE SYSTEM

The expert software for structural analysis and design optimization must mimic the activities of a typical aerospace structural design process and meet the functional specifications, described in Section 3.2. Since the activities involve a number of different types of decision areas and technologies, the expert software needs to consist of a number of cooperating expert systems with their own areas of speciality. The architecture for the expert software is shown in Figure 3-2.

An object-oriented representation of knowledge for maximum conceptual economy was used, in conjunction with a rule-based approach. The object-oriented approach promotes efficient handling of the problem data by allowing knowledge to be encapsulated in objects and organized by defining relations between objects. An Object Representation Language (ORL) is implemented as a tool for building and manipulating the object base. The associated rule-based system is used to simulate reasoning for finite element analysis and structural design. This is discussed in detail in the next chapter.

Since the essential building blocks of the expert software, including the user interface, data bases and knowledge modules (including rule sets), existing procedural modules, are almost independent of each other, a modular program design was used. As new knowledge becomes available, or radical changes occur in the design paradigm, the individual building blocks can be modified or replaced without affecting the other pieces of the software significantly.

The expert software for structural analysis and design optimization utilized pre-existing SAT computer programs for its analytical, design optimization, and other procedural aspects. These procedural programs have been developed over the years by SAT engineers. The loosely-linked design permits the use of this pre-existing software on an independent stand-alone basis.

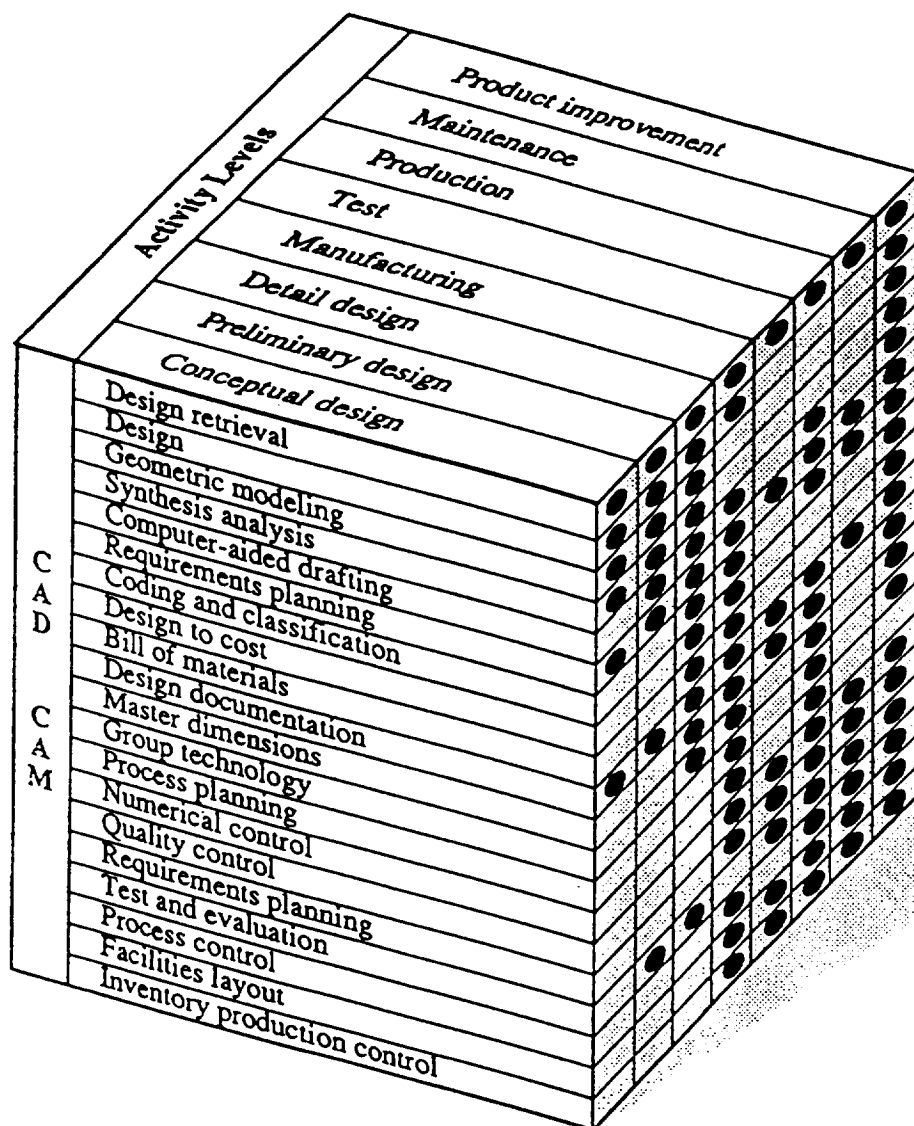


FIGURE 3-1: ACTIVITY LEVELS IN A STANDARD ENGINEERING PROJECT IN THE AEROSPACE INDUSTRY

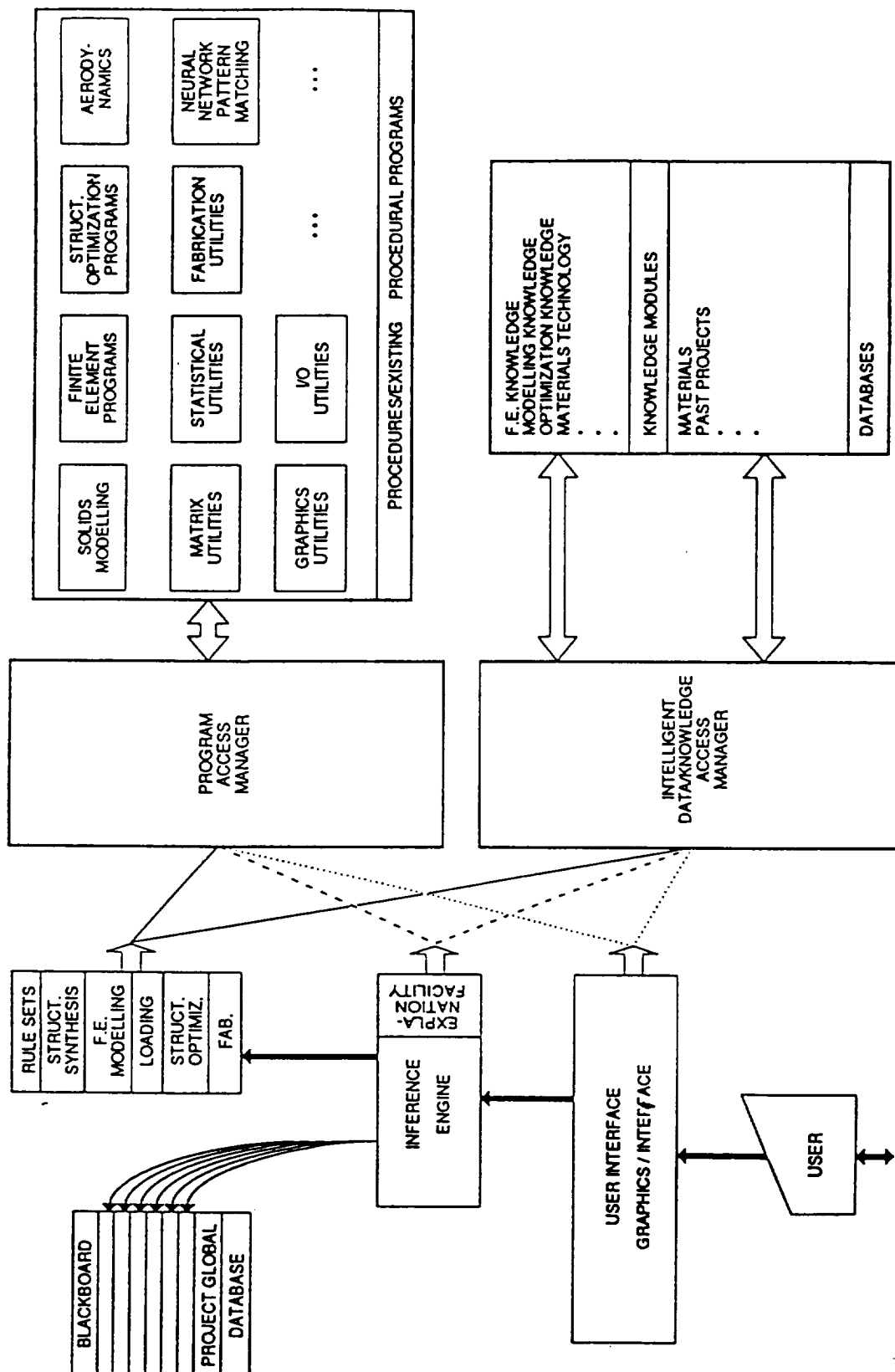


FIGURE 3-2: THE ARCHITECTURE OF THE EXPERT SOFTWARE

4. OBJECT-ORIENTED KNOWLEDGE REPRESENTATION SCHEME

4.1 INTRODUCTION

An object-oriented knowledge representation scheme has been used for the development of the knowledge base for the integrated software developed in this project. The use of object-oriented approach has promoted efficient handling of the problem data by allowing knowledge to be encapsulated in objects and organized by defining relationships between the objects. An Object Representation Language (ORL) has been implemented as a tool for building and manipulating the object base. Rule-based knowledge representation is then used to simulate engineering design reasoning. In general, the motivation for development and use of this approach for AutoDesign can be summarized as follows:

- The limited expressiveness of rule-based (only) knowledge representation scheme, especially in engineering domains,
- the inability to build large, efficient, and comprehensive expert systems consisting of thousand of rules,
- the need to effectively store knowledge (i.e. acquired from the user, data bases, or inferred by a rule set) for later use, and
- the desire to have a common environment that could link expert systems with existing data bases and procedural programs.

Even in the preliminary stages of the development of an expert system for structural/mechanical design (Ref. 94), it was realized that a system with a minimum of usefulness could be comprised of thousands of rules. This fact introduced some concerns with respect to hardware and software limitations and the practicality of maintaining such an extensive knowledge base. One of the most powerful uses of this enhancement is the ability to chain rules sets. A large set of rules can be decomposed into smaller sets which reason about specific subproblems. For example, a rule could state that if a certain piece of knowledge is unknown, then load another rule set that will infer that data. The original rule set can put itself in queue to return and continue processing, transparent to the user. Also, previously autonomous expert systems can now share data through common objects and communicate with each other through the ORL queries. As illustrated in figure 4-1, a very large network of rule sets can be developed giving the illusion of a large expert system, when in fact, only a small set of rules are being processed at any one time. This capability becomes especially important on a personal or desktop computer

platform. Developing, modifying, updating and verifying knowledge bases for large applications is a less formidable task when small rule sets can be edited and tested independent of the entire application.

Another advantage realized from this enhancement is that rule sets shrink considerably. This is primarily because rules for handling user queries and checking user responses are handled by the ORL. Rule sets need only contain rules for ORL queries, the actual problem solving rules and those rules that report the results. An existing set of rules can easily be modified to take advantage of the ORL capabilities. In the effort described herein, the ORL has been linked with the CLIPS rule-based system (Ref. 95).

Disk storage of knowledge has proven to be very useful also. In the scheme developed herein, a rule set is invoked in the context of a project. Objects are first searched for in a project specific location and then in a global storage area. In a run-time environment, modifications to the object base are only specific to a particular project. This context sensitivity allows the user to examine the effect of various responses on the recommendations or findings of an expert systems by simply changing contexts.

The ultimate intention of this effort has been to develop a fully integrated environment in which the same ORL query initiated from a rule can not only query the user but also result in a query to an existing data base or the invoking of a procedural program. The details of where the information should be retrieved would be specified as the object base is developed through the use of property metaslots. Optimally, this integration should be seamless to the user and function efficiently in a networked environment.

In the next section, the use of the ORL and the object-oriented knowledge representation scheme to build practical expert systems is discussed and demonstrated.

4.2 USE OF THE ORL

The ORL consists of a concise set of functions for building and maintaining an object base. One of the main goals in the development was to keep the use of the ORL as simple as possible so that engineers, without extensive computer programming experience, could develop knowledge bases and, furthermore, that non-experts could easily utilize the resulting expert systems.

The type of commands available include those for file operations, building and displaying classes and objects, querying and asserting property values, editing the object base and an interface to the usual CLIPS (Ref. 96) command line. The file operations allow the user to set the current project, save and

load objects to and from disk, reset memory resident object properties to unknown or to clear memory completely. Note that when running a rule set, objects are automatically loaded as needed but must be saved explicitly to permanently store any changes made by the rules.

Command line functions for building and modifying the object base include making classes and objects, making an instance of a class, copying objects, or adding and removing properties and relationships. Menu-oriented editors are available for specific modifications such as changing the name or type of a property or defining metaslots.

To access ORL commands from a rule, the developer uses the "ORL" function as the first item in the right hand side pattern. The remainder of the pattern is precisely the ORL command line function and arguments. For example, to save an object to disk from a rule, one would write:

```
{ORL save<object name>}
```

As in CLIPS, several destructive functions are disallowed from within a rule.

4.2.1 Classes and Objects

Classes and objects are the basic structures of the knowledge representation scheme. They contain descriptive properties and relationships to other classes and objects. When a property value is required in a rule set, the class or object must be queried for that specific property's value(s). Queries to classes and objects only differ in that a query to a class results in all the instances of that class being queried. In general, an ORL query from a rule takes the form:

```
{ORL get <class/object name> <property name(s)>}
```

and results in asserted facts of the form:

```
{<object name> <property name> <value> {certainty}}
```

Qualifiers for the queries such as less-than, greater-than, or equal-to need to be implemented for fully functional querying; however, these types of tests are currently available in CLIPS which accounts for their low priority in the development.

In the same way, permanent assertions to the object base take the form:

```
{ORL assert<object name> <property name> <value> {certainty}}
```

and result in the fact:

{<object name> <property name> <value> {certainty}}

Other queries return the instances of a class or related parts of an object. For example, to find out the instances of a class, the query would be:

{ORL get instances<class name>}

and would return facts as:

{<class name> instance <object name>}

which could be matched on the left hand side of a rule for deleting instances of a class.

4.2.2 Properties and Metaslots

Properties (often called "Attributes" in similar schemes) are the mechanism by which classes and objects are described. They simply hold one or more values as they are asserted. Currently, a property may be of type integer, float, text, or boolean. A property will automatically handle the checking of user responses and build the appropriate CLIPS facts as values are assigned.

Defining a metaslot for a property adds a considerable amount of versatility. First, a metaslot can be used to put constraints on the values that a property can hold by specifying a list of allowable values or a range of numeric values. Other useful features include assigning initial and default values for the property and defining the prompt displayed to the user.

Possibly, the most powerful feature of a metaslot is the ability to define a search strategy with the "Order of Sources." The USER is the default source for information when a property value is queried. Alternatively, the knowledge base developer may wish the property to assume the initial value when queried for the first time or the default value if the user responds unknown to a query. Also, it may be desirable to query an existing data base or invoke a procedural program to generate data. These facilities may lessen the need for user interaction when the level of knowledge of user may be in question.

4.2.3 Relationships

Relationships allow properties to be inherited by related classes and objects. The most common type are the *instance* and *instance of* relationships between a class and its instance. When an instance of a class is created, the

relationships between them are automatically created so that the new object can inherit properties in the class hierarchy. Other types include *is a* and *subclass* relationships between classes (e.g., *Jet is an Airplane*, *Airplane has subclass Jet*) and *part of* and *subobject* relationships between objects (e.g., *wing-x is part of airplane-y*, *airplane-y has subobject wing-x*).

As mentioned earlier, the relationships come into play when the classes and objects are queried. If a class is queried for a property value, it will automatically pass the query on to its instances. Similarly, if an object is queried for a property value which it doesn't have, it may pass the query on to related object according to the current inheritance protocol. The relationship capability promotes efficient handling of data by eliminating unnecessary redundancy.

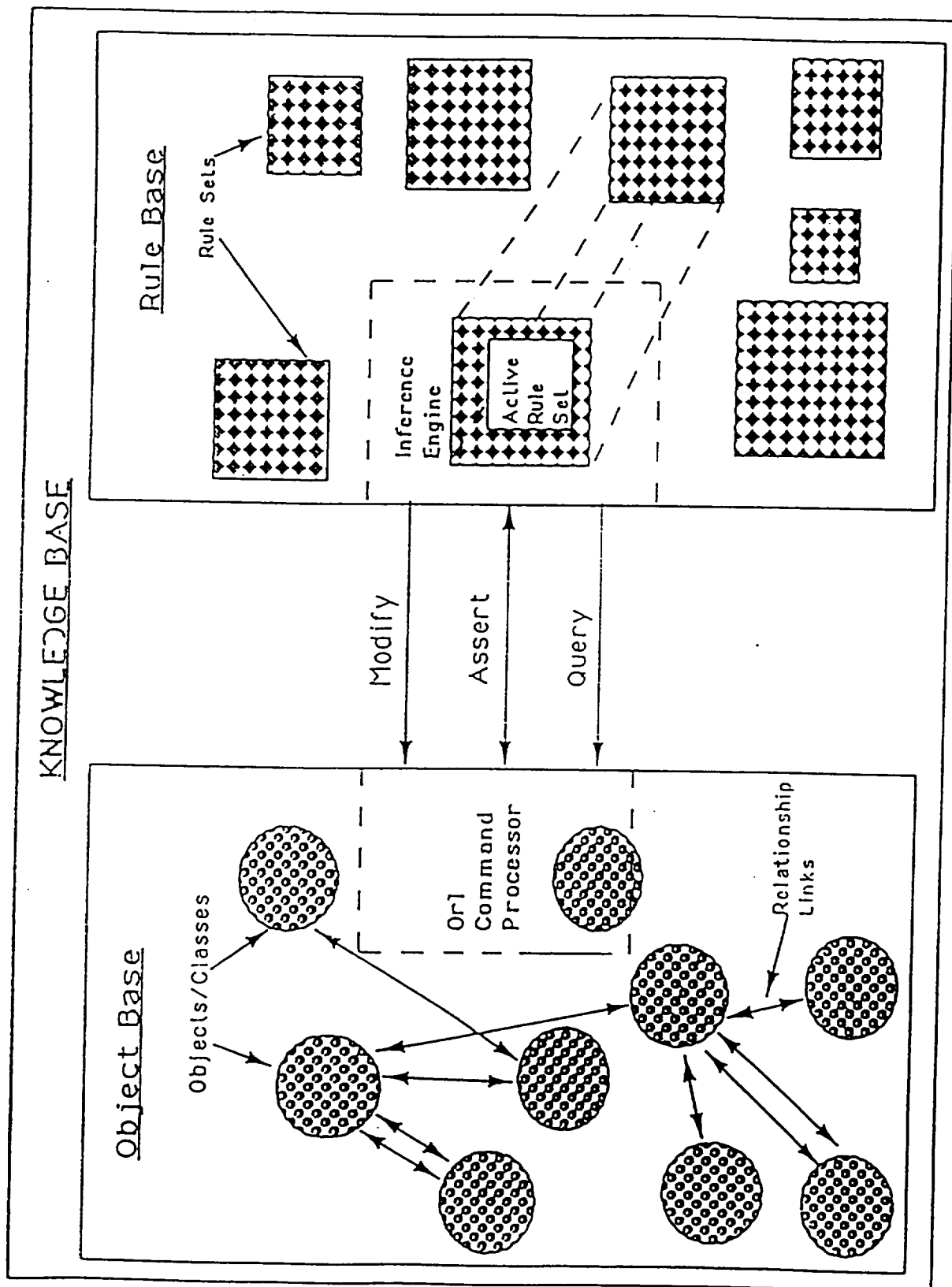


FIGURE 4-1 THE OBJECT-ORIENTED/RULE-BASED KNOWLEDGE BASE DEVELOPMENT SYSTEM

5. AUTODESIGN™ AND ITS EXPERT SYSTEM MODULES

5.1 INTRODUCTION

As discussed previously, the engineering design process involves a number of tasks requiring different types of technologies, expertise and processing. Some of these tasks are algorithmic or procedural in nature; others are heuristic and judgmental in nature. They also involve manipulation of large amounts of relevant and partially relevant data from which complex inferences must be derived.

The main objective of the research and development effort, which has culminated into AutoDesign™, has been to improve the productivity of mechanical, aerospace and structural engineers, reducing the possibility of human errors, accelerate the design process, and thus significantly reduce time to product completion in a concurrent manufacturing environment.

Currently, the design/optimization and finite element analyses are not very tightly integrated; the existing finite element analysis software are extremely difficult to learn and use (especially by engineers who are not experts in this field). The software AutoDesign™, with its built-in expertise in the form of "on-line" expert advisors, can be quickly learned and used by practically any design engineer to produce error-free, reliable and cost-effective designs in short periods of time.

A flow chart showing the various components of AutoDesign™ is shown in Figure 5-1.

5.2 EXPERT ADVISOR MODULES

The following expert advice modules are currently available:

Problem Strategy Advisors

Analysis Planning Advisor

Analysis Setup Advisors

Substructuring Advisor
Mesh refinement Advisor
Element Selection Advisor
Dynamic Modeling Advisor

Design Setup Advisors

- Constraint Type Selection Advisor
- Constraint Locations Advisor
- Design Variable Advisor
- Algorithm Selection Advisor
- Problem Simplification Advisor
- Interactive Design Advisor

These modules are described below in more detail.

Analysis Planning Advisor

This module advises the user regarding the analysis procedure to be used. Based on a variety of considerations, such as the goal of the project (e.g., preliminary design or final design, design modifications, confirmatory analysis, etc.), the type of structure and its potential behavior, given the specified geometry and loadings, the advisor recommends the type of analysis procedure (e.g., linear static, detailed dynamic time history, global dynamic/local static, global displacement/local detailed stress analysis, etc.)

Substructuring Advisor

This module advises the user whether the structure should be divided into substructures for analysis. Based on variety of considerations, such as the goal of the project, the type of structure and its size, structural geometry, type and application mode of loading, type of computer hardware, available resources, and connectivity of the various segments of the structure, the advisor recommends whether substructuring should be performed.

Mesh Refinement Advisor

This module advises the user regarding the refinement of the finite element mesh that should be used for the analysis. Based on a variety of considerations, such as the goal of the project, type of analysis, type of structure and structural system, structural geometry, openings and other features, severity of changes in thicknesses and other properties and available resources, the advisor recommends the preferable level of refinement for the finite element mesh.

Element Selection Advisor

This module advises the user regarding the type of finite element that should be used for the analysis. Based on a variety of considerations, such as the goal of the project, structure system type, structure dimensionality, type and

direction of the loading, interruptions in support continuity and interruptions in thickness, the advisor recommends the type of finite element that should be used for the analysis.

Dynamic Modeling Advisor

This module advises the user regarding dynamic modeling, e.g., how the masses should be incorporated (whether a lumped mass approach should be used; if so, how many masses should be used and where they should be lumped). Based on a variety of considerations, such as the type of structure, structure dimensionality, type of potential structural behavior, type and application mode of loading, type of analysis, type of computer hardware to be used, the advisor recommends the approach for the dynamic modeling.

Constraint Type Selection Advisor

This module assists the user in the selection of the constraint type to be used for the current design optimization problem, e.g. stress, displacement and/or frequency. Based on a variety of considerations, such as the type of structural system, dimensionality of the structure, unsupported length and type of loadings, the advisor provides recommendation regarding the type of constraint to be used.

Constraint Locations Advisor

This module assists the user in the selection of the locations where the constraints should be applied. Based on a variety of considerations, such as the goal of the project, dimensionality of the structure, support conditions of the structure, constraint types used, structural interruption, loading applications and directions, the advisor provides recommendation regarding the locations for constraint applications.

Design Variables Advisor

This module assists the user in determining the type and number of design variables for the current design problem. Based on a variety of considerations, such as the goal of the project, availability of the finite element model, types of finite elements being used, dimensionality of the structure, type of structure assembly, unsupported length, structural interruptions, loading types, the advisor provides recommendations regarding the type and number of design variables for the design problem.

Problem Simplification Advisor

This module assists the user in the simplification of the Optimization Problem. Based on a variety of considerations, such as the goal of the project, type of structure, types of constraints, structural behavior, structural dimensionality, magnitude of loadings, types of supports, complexity of the structure, unsupported length and other properties, the advisor provides recommendations about the order in which the displacement, stress and frequency constraints may be applied, as well as the approach for reduction in the number of constraints and design variables to simplify the optimization problem without major loss of accuracy.

Algorithm Selection Advisor

This module advises the user in the selection of the optimization algorithm. Based on a variety of considerations, such as goal of the project, constraints being used, type of mode (interactive, batch), etc., the advisor provides recommendations on which algorithm to use (e.g., SQP, CFB, Hybrid, etc.), and which subalgorithm to use (e.g., R1, R2, R3, R4), etc.

Interactive Design Advisor

This module advises the user, when invoked at any step during the optimization process, whether the optimization algorithm or subalgorithm should be changed during the subsequent iterations, and what parameters (e.g. tolerances, convergences criteria, etc.) should be changed and how? Based on a variety of considerations, such the type of algorithm currently being used, the subproblem being used, percent violation of the current design, these recommendations are provided for interactive optimization.

5.3 SAMPLE DIALOGUE

A sample dialogue between the software and the user is presented below for the Analysis Planning Advisor. Similar dialogues are carried out for other modules.

Analysis Planning Advisor

This Expert System provides advice concerning the type(s) of analysis that should be performed for the current stage of design.

Would you like to continue?

Yes/No value (1 = yes, 0 = no);

The user may answer 1.

What loadings are imposed on the structure?

The following choices are available:

- Pressure
- Thermal
- Dynamic-Time-History
- Aeroelastic
- Aerodynamic
- Impactive
- Impulsive

The user may select one or more of the above loadings.

Describe the behavior of the structure.

The following choices are available:

- Axial
- Membrane
- Shear
- Flexural
- Torsional
- Static
- Dynamic
- Global Modes Dominant
- Local Modes Dominant
- Mixed

The user may select one or more of the above choices.

The dialogues, as shown above, and similar dialogues for other modules, are used to get the information from the user about his problem.

Based on the answers to the questions asked, a number of rules are fired and recommendations are provided.

For example, for the **Analysis Planning Advisor**, the following recommendations may be displayed:

- Linear
- Static

In addition to the recommendations, the certainty associated with the recommendations, may be displayed. A summary of the objects and the answers supplied by this user to the questions asked by the expert advisor module may also be provided.

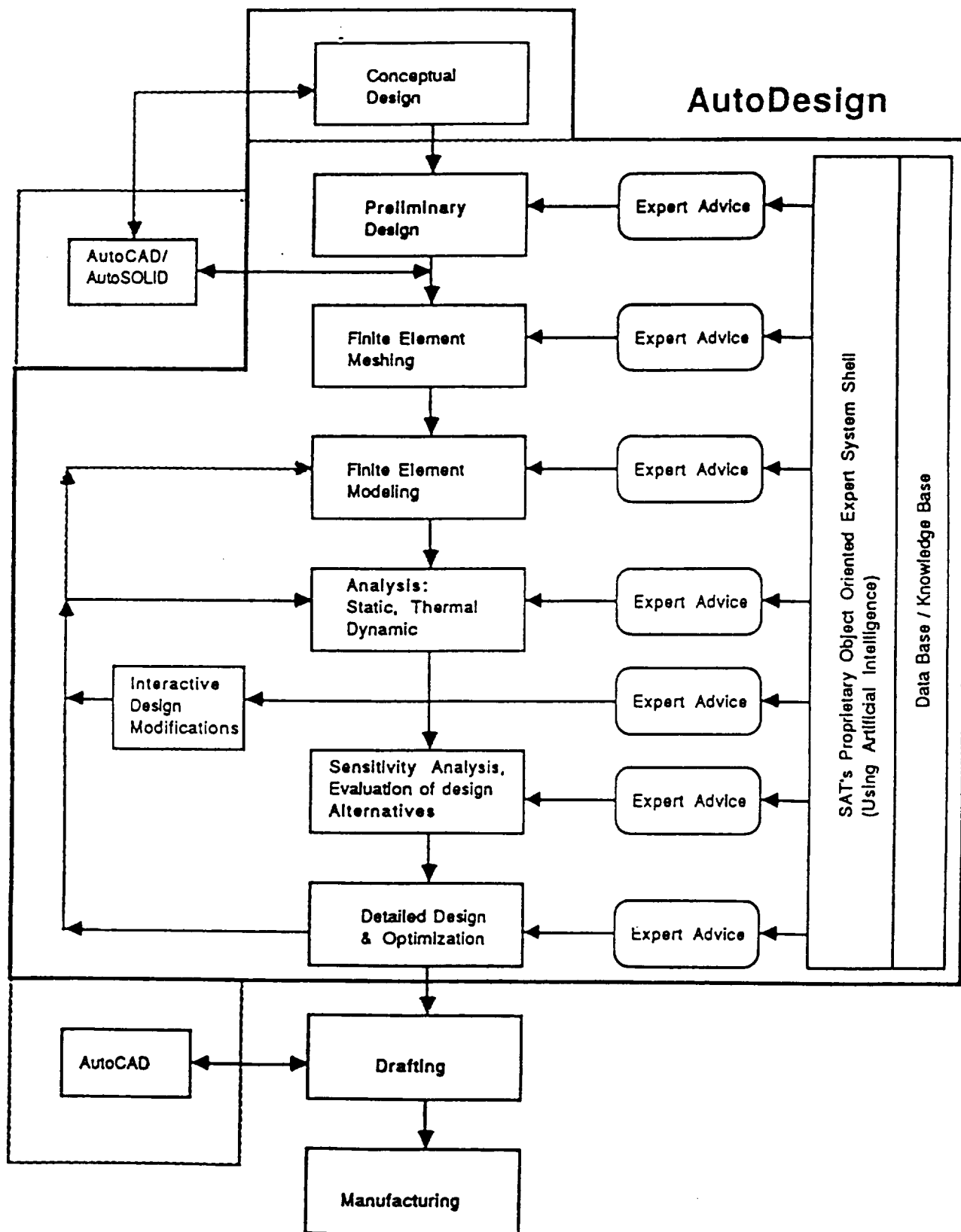


FIGURE 5-1 FLOWCHART FOR THE SOFTWARE, AUTODESIGN™

6. CONCLUSIONS

This report presented the results of a research study on the development of an integrated expert software for structural analysis and design optimization of aerospace structures. An Object Representation Language (ORL), in conjunction with a rule-based system, was developed first, which was then used for the development of the expert system modules of the integrated software. These expert system modules were developed in conjunction with a procedural program, AutoDesign™ (developed in-house at SAT, Inc.). The expert system modules so developed will provide assistance to the user in the development of the structural model, selection of finite element types, selection of structural analysis procedures, selection of design optimization algorithms/procedures, selection of design variables, selection of constraints and their locations, and various other tasks associated with structural analysis and design optimization. The basic idea is to allow the user to make better analysis and design decisions to improve his productivity and reduce time to completion of design.

An extensive literative survey and feasibility study was performed in Phase I, and an architecture and conceptual design for the integrated software was developed. In Phase II, a detailed development of such an integrated software was carried out.

As discussed in the report, an object-oriented knowledge representation scheme, in conjunction with a rule-based system, was used. An Object Representation Language (ORL) was developed for this purpose. It was concluded that this approach was highly efficient and effective for development of knowledge based expert systems for engineering applications in general, and applications to structural analysis and design in particular. This is because the object-oriented approach used herein has significantly more expressiveness than a rule-based only approach used more commonly; it provides the ability to build large, efficient and comprehensive expert systems through the development of many small rule sets communicating through the object-oriented environment; it provides ability to store knowledge in an efficient manner, as well as the capability to rapidly develop knowledge and easily modify and maintain it.

Almost a dozen expert system modules were developed using this approach, and were applied to various practical problems, in conjunction with the procedural structural analysis and design optimization modules, in the form of the integrated software, AutoDesign™. The use of this expert software was found to have significantly improved the ease of modeling and performing structural analysis and design optimization. The software was beta-tested at

several companies. It was discovered that design engineers who did not have extensive background in finite element structural analysis and design optimization found it easy to develop finite element models, perform structural static and dynamic analyses, set up a design optimization problem, and carry out integrated analysis and design for a variety of aerospace and mechanical problems.

In addition, selected users were also allowed to use the ORL/AI shell to develop their own knowledge-based expert systems for their specific applications, to be used in conjunction with the expert systems available in the integrated software. In general, these users found the ability to add their own knowledge to the integrated software to be fascinating and useful.

The following conclusions were reached based on the experience gained by us on this project, as well as the feed back obtained from the users of this integrated expert software.

- The development of knowledge base in AutoDesign™ is going to be a continuing process. The basic knowledge associated with finite element structural analysis and design optimization has been incorporated. This is oriented towards engineers who are not experts in finite element analysis and design optimization. However, this knowledge would need to be continuously updated, especially for specific problems and applications.
- The ORL/AI shell can be very effectively and efficiently used by a variety of companies to add their own knowledge to the system based on their "design culture" and their own special way of performing structural analysis and design optimization. This approach can be also be very useful in acquiring knowledge from senior design engineers at the time of their retirement.
- The expert systems should be more integrated with the procedural finite element structural analysis and design optimization, and more automation should be provided so that the recommendations of the expert advisors are automatically carried out by the procedural software. It must be pointed out, though, that there is difference of opinion in the user community regarding such automation. It is the opinion of some users that the user must have control on the acceptance or rejection of the recommendations of the expert advisors. Others feel that the recommendations of the expert advisors should be automatically implemented by the software.
- The ORL/AI shell needs additional utilities to improve its ease of use, including easy graphical display of objects and rules, and additional windows, pull-down menus and icons in the user interface.

REFERENCES

1. Alexander, C., *Notes on Synthesis of Form*, Harvard University Press, Massachusetts, 1964.
2. Lawson, B., *How Designers Think*, The Architectural Press Ltd., London, 1980.
3. Mostow, J., "Toward Better Models of the Design Process," *AI Magazine*, Vol. VI, pp 44-56, 1985.
4. Simon, J. A., *The Sciences of the Artificial*, MIT press, Cambridge, MA, USA, 1969.
5. Bennett, J.S. and Engelmores, R.S., SAGON: A Knowledge-Based Consultant for Structural Analysis," *Proceedings, Sixth IJCAI*, pp. 47-49, August 20-23, 1979.
6. Maher, M. L., and Fenves, S. J., "HI-RISE: an Expert System For The Preliminary Structural Design of High Rise Buildings," in *Knowledge Engineering in Computer-Aided Design*, Gero, J., Ed., North Holland, pp. 125-146, 1985, [For a detailed description of HI-RISE, see Report No. R-85-146, Dept. of Civil Engineering, C-MU, PA 15213].
7. Sriram, D., *Implementation Details of ALL-RISE*, unpublished, 1986, [Civil Engineering Report, C-MU, Pittsburgh, PA 15213].
8. Dym, C.L., Ed., *Applications of Knowledge-Based Systems to Engineering Analysis and Design*, American Society of Mechanical Engineers, 345 East 47th St., NY 10017, 1985.
9. Gero, J., Ed., *Knowledge Engineering in Computer Aided Design*, North-Holland Publishing company, New York, 1985.
10. Brackman, R., and J. Schmolze, "An Overview of the KL-ONE Knowledge Representation system," *Cognitive Science* 9(2), 1985.
11. Bobrow, D., and T. Winograd, "An Overview of KRL, a Knowledge Representation Language," *Cognitive Science* 1(1): 3-46 (1977).
12. Stefik, M., Bobrow, D., Mittal, S., and Conway, L., "Knowledge Programming in LOOPS," *AI Magazine* 4(3): 3-13 (1983).

13. Woods, W., "What's in a Link?", In *Representation and Understanding*, edited by D. Bobrow and A. Collins. New York, N.Y.; Academic Press, 1975.
14. Sriram, D. (Editor), *Knowledge-Based Expert Systems for Engineering*, unpublished, 1986 [Forthcoming from CML Publications, UK].
15. Cabonell, J.G., Larkin, J. H., and Reif, F., *Towards a General Scientific Reasoning Engine*, Technical Report, Dept. of computer Science, Carnegie-Mellon University, 1983.
16. Maher, M. L., Sriram, D., and Fenves, S. J., "Tools and Techniques for Knowledge-based Expert Systems for Engineering Design," *Advances in Engineering Software*, October 1984.
17. Norabhoompipat, T. and Fenves, S.J., *An Information Processing Model of Civil Engineering Design Systems*, Technical Report, R-78-110, Department of Civil Engineering, Carnegie-Mellon University, August 1978.
18. Erman, L. D. and Lesser, V. R., "A Multi-Level Organization for Problem Solving Using Many, Diverse, Cooperating Sources of Knowledge," *Proceedings*, Fourth IJCAI, pp. 483-490, 1975.
19. Erman, L. D., Hayes-Roth, F., Lesser, V. R., and Reddy, D. R., "The Hearsay-II Speech Understanding System: Integrating Knowledge to Resolve Uncertainty," *Computing Surveys*, Vol. 12, No. 2, pp. 213-253, 1980.
20. Rehak, D. R., Howard, H. C., and Sriram, D., "Architecture of an Integrated Knowledge-Based Environment for Structural Engineering Applications," in *Knowledge Engineering in Computer-Aided Design*, Gero, J., Ed., North Holland, 1985.
21. Hayes-Roth, B., *The Blackboard Architecture: A General Framework for Problem Solving*, HPP 83-30, Dept. Computer Science, Stanford University, May 1983.
22. Amarel, S., "Basic Themes and Problems in Current AI Research", *Proceedings of the Fourth Annual AIM Workshop*, Ceilsielske, V.B., Ed., Rutgers University, p. 28-46, June 1978.

23. Blazer, R., Erman, L.D., London, P., and Williams, C., "Hearsay-III: A domain Independent Framework for Expert Systems," *Proceedings of the First Annual National Conference on Artificial Intelligence (1st NCAI)*., pp. 108-110, 1980.
24. Banares, R., Westerberg A., Ko, E., and Rychener, M., "DECADE: A Hybrid Expert System for Catalyst Selection," *AIChE Annual Meeting*, Chicago, AIChE, September 1985.
25. Brown, D. and Chandrasekaran, B., "Expert Systems for a Class of Mechanical Design Activity" in *Knowledge Engineering Computer-Aided Design*, Gero, J., Ed., North Holland, pp. 259-290, 1985.
26. Bushnell, M., ULYSSES: *An Expert System Based VLSI Design Environment*, unpublished Ph.D. Dissertation, Department of electrical Engineering, Carnegie-Mellon University, 1986.
27. Robson, D., "Object-Oriented Software Systems," *BYTE*, pp. 74-86, August 1981.
28. Cabonell, J. G., *Derivation Analogy: A Theory of Reconstructive Problem Solving Expertise Acquisition*, CMU-CS-85-115, Dept. Computer Science, Carnegie-Mellon University, March 1985.
29. Corkill, D. and Lesser, V. R., "A Goal-Directed HEARSAY-II Architecture: Unifying Data-directed and Goal-directed Control," *Proceedings of Second NCAI*, Pittsburgh, AAAI, August 1982.
30. de Kleer, J., "Choices Without Backtracking," *Proceedings of the 4th NCAI*, Texas AAAI, August 1984.
31. Dixon, J. and Simmons M., "VEXPERT: An Expert System for V-Belt Designs," *Computers in Engineering - Advanced Automation: 1984 and Beyond*, Gruver, W.A., Ed., A.S.M.E., 1984.
32. Doyle, J. "A Truth Maintenance System," *Artificial Intelligence*, Vol. 12, pp. 231-272, 1979.
33. Dyer, M. and Flower, M., "EDISON: An Engineering Design Invention System Operating naively," *Proceedings of AI in Engineering*, Southampton, UK, CML Publications, April 1986, in press.
34. Eastman, C.M., "Database Facilities for Engineering Design," *Proceedings of the IEEE*, Vol. 69, No. 10, pp. 1249-1263, October 1981.

35. Fenves, S. J. and Garrett, J., "Knowledge-Based Standards Processing," To appear in the International Journal for AI in Engineering, July, 1986.
36. Hayes-Roth, B., Waterman, D., and Lenat, D., Eds., *Building Expert Systems*, Addison-Wesley Publishing Company, Inc., 1983.
37. Hayes-Roth, B., "A Blackboard Architecture for Control," *Artificial Intelligence*, Vol. 26, No. 3, pp. 251-321, 1985, [Also Technical Report No. HPP 83-38, Stanford University].
38. Howard, H.C., *Integrating Knowledge-Based Systems with Database Management Systems for Structural Engineering Applications*, unpublished, Dept. of Civil Engineering, Carnegie-Mellon University, April 1984, [Ph.D. Thesis, Department of Civil Engineering, Carnegie-Mellon University. See also Special Issue of CAD, November 1985].
39. Howe, A., Dixon, J., Cohen, P., and Simmons, M., "DOMINIC: A Domain-Independent Program for Mechanical Engineering Design," to appear in The International Journal for AI in Engineering, July, 1986.
40. Kedar-Cabelli, S., Purpose-Directed Analogy, ML-TR-1, Lab. for Computer Science Research, Rutgers University, July 1985.
41. Lin, T. Y. and Stotesbury, S. D., *Structural Concepts and Systems for Architects and Engineers*, John Wiley & Sons, 1981.
42. McDermott, J., R1: A Rule-Based Configurer of Computer Systems, Technical Report CMU-CS-80-119, Carnegie-Mellon University, 1980.
43. Mitchell, T., Steinberg, L., Kedar-Cabelli, S., Kelly, V., Shulman, J., and Weinrich, T., "An Intelligent Aid for Circuit Redesign," *Proceedings 3rd NCAI*, Washington, D. C., AAAI, pp. 274-278, August 1983.
44. Mitchell, T., Steinberg, L. and Shulman, "A Knowledge-Based Approach to VLSI Design," *IEEE Transactions on PAMI*, pp. 502-510, September 1985.
45. Mittal, S., Dym, C., and Morjaria, M., "PRIDE: An Expert System for the Design of Paper Handling Systems," in *Applications of Knowledge-Based Systems to Engineering Analysis and Design*, Dym, C., Ed., American Society of Mechanical Engineers, pp. 99-116, 1985.
46. Moore, R. L., Howkinson, L. B., Knickerbocker, C. G. and Churchman, L. M., "A Real Time Expert System for Process Control," *Proceedings of the IEEE First Conference on AI Applications*, 1984.

47. Nakazawa, N., et al., "Interactive Design System for Structure Analysis and Strength Evaluation: HIDESS," *Computers in Engineering*, Cokonis, T. J., Ed., ASME, pp. 55-61, 1982.
48. Perrone, J. and Pilkey, W., *Structural mechanics Software Series*, Vol. 4, University Press of Virginia, University of Virginia, Charlottesville, 1982, [See other volumes in this series].
49. Prasad, B., "An Integrated System Approach to Structural Syntheses," *Eighth Conference on Electronic Computation*, Nelson Jr., J. K., Ed., pp. 155-171, February 1983, [Sponsored by the Structural Division of ASCE].
50. Rasdorf, W. J., *Structure and Integrity of a Structural Engineering Design Database*, unpublished Ph. D. Dissertation, Department of Civil Engineering, Carnegie-Mellon University, April 1982, [Available as DRC technical report no: DRC-02-14-82].
51. Smith, R. G., "STROBE: Support for Structured Object Knowledge Representation," *Proceedings Seventh IJCAI*, 1983.
52. Rogers, J. L., Feyock, S., Sobieski, J.S., "STRUCTEX-A Prototype Knowledge-Based System for Initially Configuring a Structure to Support Point Loads in Two Dimensions."
53. Zari, C., "Expert Systems and Information Retrieval: An Experiment in the Domain of Biographical Data Management," *International Journal of Man-Machine Studies*, Vol. 20, No. 1, pp. 87-106, June, 1984.
54. Kaplan, S. J., "Designing a Portable National Language Data Base Query System," *ACM Transaction on Data Base Management Systems*, Vol. 9, No. 1, pp. 1-19, March, 1984.
55. Gallaire, H., and Minker, J., Eds., *Logic and Data Bases*, Plenum Press, New York, 1978.
56. Gallaire, H., and Minker, J., and Nicolas, J.M., Eds., *Advances in Data Base Theory*, Vol. 1, Plenum Press, New York, 1981.
57. Gallaire, H., and Minker, J., and Nicolas, J.M., "Logic and Databases: A Deductive Approach," *ACM Computing Surveys*, Vol. 16, No. 2, pp. 153-185, June 1984.
58. Kellogg, C., Klahr, P., and Travis, L., "Deductive Planning and Pathfinding for Relational Databases," in *Logic and Database*, H. Gallaire, J. Minker, and J.M. Nicholas, Eds., Plenum Press, New York, pp. 179-200, 1978.

59. Kellogg, C., Klahr, P., And Travis, L., "Reasoning with Data in a Deductively Augmented Database System," in *Advances in Database Theory*, Vol. 1, H. Gallaire, J. Minker, Eds., Plenum Press, New York, pp. 261-295, 1981.
60. Kellogg, C., "Knowledge Management: A Practical Amalgam of Knowledge and Data Base Technology," *Proceedings, National Conference on Artificial Intelligence*, Pittsburgh, PA, pp. 306-309, August, 1982.
61. Kellogg, C., "The Intelligent Assistants for Knowledge and Information Resources Management," *Proceeding of the International Joint Conference on Artificial Intelligence*, Karlsruhe, West Germany, August, 1983.
62. Kellogg, C., "The Transition from Data Management to Knowledge Management," *Proceedings, International Conference on Data Engineering*, Los Angeles, California, Institute of Electrical and Electronics Engineers, IEEE Computer Society Press, pp. 467-472, April, 1984.
63. Klahr, P., "Planning Techniques for Rule Selection in Deductive Question-Answering," in *Pattern-Directed Inference Systems*, D.A. Waterman and F. Hayes-Roth, Eds., Academic Press, New York, 1978.
64. Jakobson, G., Lafond, C., Nyberg, E., and Paitiesky-Shapiro, G, "An Intelligent Data Base Assistant," *IEEE Expert*, Vol. 1, No. 2, pp. 65-78, 1986.
65. Blum, R.L., "Discovery, Confirmation, and Incorporation of Causal Relationships from a Large Time-Oriented Clinical Data Base: The RX Project," *Computers and Biomedical Research*, Vol. 15, No. 2, pp. 164-187, April 1982.
66. Blum, R.L., "Discovery, Confirmation, and Incorporation of Causal Relationships from a Large Time-Oriented Clinical Data Base: An Overview of the RX Project," *Proceedings of the National Conference on Artificial Intelligence*, Pittsburgh, Pennsylvania, American Association for Artificial Intelligence, pp. 355-357, August, 1982.
67. Blum, R.L., *Discovery, and Representation of Causal Relationships from a Large Time-Oriented Clinical Data Base: The RX Project*, unpublished Ph.D Dissertation, Stanford University, Stanford, CA 1982.

68. Kanamori, Y., Masunaga, Y., Kido, K., and Noguchi, S., "Design of a Database System for Skull Line Drawings Processing in Orthodontics Based on the Relational Model," *Medinfo 80*, D.A. Lindberg and K. Shigekoto, Eds., Third World Conference on Medical Informatics, Tokyo, International Federation for Information Processing and International Medical Informatics Association, North-Holland Publishing Co., New York, pp. 1154-1158, September-October, 1980.
69. Kanamori, Y., and Sugawara, J., "An Orthodontic Case Study Instruction System Based on a Relational Database System," *Database Engineering*, IEEE, Vol. 6, No. 4, pp. 46-49, December 1983.
70. Howard, H.C., *HICOST*, unpublished Project Report, 12-743 Expert Systems in Civil Engineering Department of Civil Engineering, Carnegie-Mellon University, December 1983.
71. Maher, M.L., and Howard, H.C., "Accessing Design Databases from Expert Systems," *ACM Sigart Newsletter*, Special Interest Group on Artificial Intelligence, Association for Computing Machinery, Spring 1985.
72. Rychener, M.D., *PSRL: An SRL-Based Production-Rule System*, DRC-05-03-84, Design Research Center, Carnegie-Mellon University, Pittsburgh, PA, December 1984.
73. Wright, J.M., and Fox, M.S., *SRL Schema Representation Language*, Technical Report, Intelligent Systems Laboratory, The Robotics Institute, Carnegie-Mellon University, December 1983.
74. Stonebraker, M., Wong, E., and Kreps, P., "The Design and Implementation of INGRES," *ACM Transactions on Database Systems*, Vol. 1, No. 3, pp. 189-222, September 1976.
75. Jarke, M., and Vassiliou, Y., "Coupling Expert Systems with Database Management Systems," *Proceedings of the NYU Symposium on AI Applications to Business*, May, 1983.
76. Jarke, M., Clifford, J., and Vassiliou, Y., "An Optimizing Prolog Front-End to A Relational Query System," *SIGMOD '84*, B. Yormark, Ed., Annual Meeting, Boston, Massachusetts, ACM SIGMOD, pp. 296-306, June, 1984.
77. Jarke, M., and Vassiliou, Y., "Databases and Expert Systems: Opportunities and Architectures for Integration," in *New Applications in Data Bases*, G. Gardarin and E. Gelenbe, Ed., Academic Press, London, pp. 185-201, 1984.

78. Vassiliou, Y., Jarke, M., and Clifford, J., "Expert Systems for Business Applications: A Research Project at New York University," *Database Engineering*, IEEE, Vol. 6, No. 4, pp. 50-55, December 1983.
79. Lopez, L.A., Elam, S.L., and Christopherson, T., "SICAD: A prototype Implementation System for CAD," *Proceedings, ASCE Third Conference on Computing in Civil Engineering*, San Diego, CA, American Society of Civil Engineers (ASCE), pp. 84-94, April 1984.
80. Taig, I.C., "Modeling for the Finite Element Method," AGARD Lecture Series No. 147 entitled "*Practical Application of Finite Element Analysis to Aircraft Structural Design*," Report No. AGARD-LS-147, August 1986.
81. Vanderplaats, G.N., "Structural Optimization Methods for Industrial Application," *Recent Trends in Aeroelasticity, Structures and Dynamics*, edited by P. Hajela, 1987.
82. Vanderplaats, G.N., "CONMIN - A FORTRAN Program for Constrained Function Minimization: User's Manual," NASA TM X-62, 282, August, 1973.
83. Schmit, L.A. and Miura, H., "A new Structural Analysis/Synthesis Capability - ACCESS1," *AIAA Journal*, Vol. 14, No. 5, May 1976, pp. 661-671.
84. Haftka, R.T. and Starnes, J.H., Jr., "Application of a Quadratic Extended Interior Penalty Function for Structural Optimization," *AIAA Journal*, Vol. 14, 1976, pp. 718-724.
85. Lasdon, L., "GRG2 User's Manual," University of Texas, Austin, Texas.
86. OPT Library User's Manuals, K. Ragsdall, University of Missouri, Columbia, Missouri.
87. OPTDES, BYU User's Manual, Design Optimization Laboratory, Brigham Young University, Provo, Utah.
88. Arora, J.S., Thanedar, P.B. and Tseng, C.H., "User's Manual for Program IDESIGN, Version 3.4," Technical Report No. ODL-85-10.
89. Vanderplaats, G.N., Sugimoto, H. and Sprague, C.M., "ADS-1: A New General-Purpose Optimization Program," *AIAA Journal*, Vol. 22, No. 10, Oct. 1984.

90. Khot, N.S., "Optimality Criteria Methods in Structural Optimization," Chapter 5, *Foundation of Structural Optimization: A Unified Approach*, edited by A.J. Morris, 1982.
91. Berke, L. and Khot, N.S., "Structural Optimization Using Optimality Criteria," NATO ASI Series, Vol. F27, *Computer-Aided Optimal Design: Structural and Mechanical Systems*, edited by C.A. Mota Soares, Heidelberg, 1987.
92. Berke, L. and Khot, N.S., "Use of Optimality Criteria methods for Large Scale Systems," AGARD Lecture Series No. 70, Structural Optimization, 1974.
93. Berke, L. and Khot, N.S., "Structural Optimization Using Optimality Criteria," Lisbon, Portugal, 1987.
94. Logie, D.S., Kamil, H. and Umaretiya, J.R., "An Application of Object-Oriented Knowledge Representation to Engineering Expert Systems", *Proceedings, Third Air Force/NASA Symposium on Recent Advances in Multidisciplinary Analysis and Optimization*, San Francisco, California, September, 1990.
95. Logie, D.S., Kamil, H., "Integration of Object-Oriented Knowledge Representation with the CLIPS Rule-Based System", *Proceedings, First CLIPS User's Group Conference*, Houston, August, 1990.
96. Kamil, H., Vaish, A.K., and Berke, L., "An Expert System for Integrated Design of Aerospace Structures", *Proceedings, Fourth International Conference on Application of Artificial Intelligence in Engineering (AIENG 89)*, Cambridge, England, July, 1989.